SEVENTH FRAMEWORK PROGRAMME
Challenge 1
Information and Communication Technologies

Trusted Architecture for Securely Shared Services

Document Type: Deliverable

Title:

Pilots Specifications and Use Case Scenarios

Work Package: WP09

Deliverable Nr: D9.1

Dissemination: PU

Preparation Date: December 29, 2009

Version: 2.1.0

Legal Notice

All information included in this document is subject to change without notice. The Members of the TAS^3 Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the TAS^3 Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

SEVENTH FRAMEWORK
PROGRAMME

The TAS^3 Consortium

	Beneficiary Name	Country	Short	Role
1	KU Leuven	BE	KUL	Coordinator
2	Synergetics NV/SA	BE	SYN	Partner
3	University of Kent	UK	KENT	Partner
4	University of Karlsruhe	DE	KARL	Partner
5	Technische Universiteit	NL	TUE	Partner
6	CNR/ISTI	IT	CNR	Partner
7	University of Koblenz-Landau	DE	UNIKOL	Partner
8	Vrije Universiteit Brussel	BE	VUB	Partner
9	University of Zaragoza	ES	UNIZAR	Partner
10	University of Nottingham	UK	NOT	Partner
11	SAP Research	DE	SAP	S&T Coord.
12	EIfEL	FR	EIF	Partner
13	Intalio	UK	INT	Partner
14	Risaris	IR	RIS	Partner
15	Kenteq	NL	KETQ	Partner
16	Oracle	UK	ORACLE	Partner
17	Custodix	BE	CUS	Partner
18	Medisoft	NL	MEDI	Partner
19	Symlabs	PT	SYM	Partner

Contributors

	Name	Organisation
1	Brecht Claerhout	Custodix
2	Sandra Winfield	University of Nottingham
3	Tom Kirkham	University of Nottingham

Contents

1 Executive Summary

The objective of WP9 'Employability and Healthcare Demonstrators' is to prove the generic applicability of the TAS^3 trust infrastructure for exchanging and managing personal information in different domains, in particular in the areas of employability and healthcare. The first iteration of 'D9.1 - Pilots Specifications and Use Case Scenarios' focused on describing the use case scenarios for the TAS^3 pilots in the healthcare and employability domains. This second iteration describes two 'integration trials' which embody the first practical (i.e. technical) steps towards establishing the TAS^3 pilots. Later iterations will document implemented pilot situations in Healthcare, and in UK and NL employability.

TAS^3 is a complex environment. Over 50 individual components are currently being defined and are still under active development (i.e. only a limited set of functionality is available to date). The actual pilot applications will rely on many functional aspects and thus a large subset of TAS^3 (technical) components. In order to establish the robustness of the TAS^3 software, a number of integration trials are planned in which the conditions and story-boards of the pilots are approximated.

Integrating at such an early stage in the development process provides valuable feedback for component developers regarding the quality of their components (e.g. bugs, stability, platform dependence...). In addition, it allows decisions on integration strategy (e.g. interfaces, component division...) to be evaluated at an early stage, keeping the effort required for corrective measures to a minimum. In fact this approach can be considered as an informal form of the continuous integration paradigm.

Integration is being approached from two different angles in WP9: 'top-down', meaning that TAS^3 components are integrated into an existing system; and 'bottom-up', in which a use case scenario is implemented from scratch. The former is being used in the healthcare domain, the latter in the employability domain.

The two working TAS^3-enabled systems (integration trials) will be demonstrated during the March 2010 review (i.e. live demonstration of systems). The outcomes of the integration trial will be reported in 'D9.2 - Pilot evaluation report'.

2 Introduction

2.1 'Integration Trials' and 'Pilots'

As the TAS³ environment consists of over 50 components, it was decided that before any actual piloting could take place there would need to be an extensive integration trial using dummies of live systems to test the robustness of the architecture.

The decision was taken to pursue this in two different ways: using a 'top-down' approach, which means integrating TAS³ components into an existing system; and using a 'bottom-up' approach, in which a use case scenario is implemented from scratch (Table 1). In fact the latter is not a completely bottom-up approach as the implementation is based as much as possible on readily available (open source) components so that custom code can be reduced to a minimum.

Integration Trial	Application Domain	Legacy / Green Field	TAS3 integration
PILS/Repository integration Trial	Healthcare	Legacy / Top-down	TAS3 components are <u>replacing</u> the existing security advanced model
Student employability Trial	Employability	Green field / Bottom-up	TAS3 components are included from the <u>design stage</u> on

Table 1: Integration trials

This document describes the two integration trials: later iterations will document actual pilot situations, in Healthcare and in UK and NL employability (cf. also the first iteration of D9.1).

In both healthcare and employability environments, a number of different types of 'end-user' will be brought into contact with TAS³ (Table 2). The current integration trials focus most on the least obvious group: application developers. The integration trials focus mainly on getting TAS³ components to work effectively in combination with other software. While the project continues to keep the principle of user centricity at its heart, the evaluation of TAS³ in this initial implementation iteration will from necessity focus on technical usability and interoperability, rather than on user-friendliness and on enhancing citizen (data subject) empowerment.

Who	Healthcare	Employability
Data subjects (working with their own data)	Patients	Employees, people looking for work, students looking for work placements or part-time

		work to enhance employability
People working with data of other people	Relatives of Patients, Health Care Professionals (HCPs)	Employers, placement providers, placement co-ordinators
Application Developers	Application developers, some are aware of security technology and implementations	Application developers, typically with little awareness of security and data protection issues

Table 2: Types of TAS^3 users

It is not the objective of this document to detail TAS^3 component functionality. The components are extensively described in the other TAS^3 deliverables, in particular in 'D12.4 - TAS3 Integrated System'. The reader should refer to these documents should the functionality of the mentioned components be unclear.

3 Healthcare Integration Trial

3.1 Legacy Application: PILS / Summary Record

3.1.1 Introduction

The drivers behind the current eHealth evolutions have been thoroughly explained in D1.1. Providing 'Continuity of Care' is a cornerstone in improving quality of care and can only be delivered if a patient can be followed from cradle to grave by Health Care Professionals (HCPs) who at any given point in time have access to the relevant medical history of a patient.

Improvement of information exchange between HCPs is therefore crucial. Given the fact that medical information is inherently distributed (it is stored at the point where it is 'produced'), one way of tackling this problem is via virtual federation of electronic patient information: initially through federating hospital information stores (obvious data sources with in-house IT capability on relatively large populations), but also summary records, primary care physician records, homecare and eventually even Personal Health Records (PHRs). It is clear that ensuring confidentiality is fundamental in this evolution, i.e. dealing with security has to be a major part of this work.

The trial scenario is staged in the Belgian healthcare environment (which is comparable to many other EU countries). In Belgium, a number of hospitals provide access to their Hospital Information System (HIS) or a results server (see below) to professionals (essentially GPs). Patient access is currently very rare (except possibly for downloading medical images). Exchange of medical data is still mostly based on 'documents', which is reflected in many electronic data formats and in the use of result servers which contain episode reports (such as a lab outcome, a radiology report, a medical consultation report, ...). In Belgium, the KMEHR (Kind Messages for Electronic Healthcare Record) XML standard is used for exchange of medical information[1].

Custodix has previously implemented a simple KMEHR repository service as part of an internal project focussing on application security. This repository is capable of storing medical results in KMEHR format and can therefore act as hospital result server, summary record service, etc.

Federated access to different instantiations of this repository service is provided through a Patient Information Location Service (PILS). This information locator is not an index server, as commonly proposed[2] in healthcare data federation, but rather a search engine supporting distributed searching. The focus of the implementation is on security and the enforcement of complex access policies. The main goal of this KISS ('Keep it Simple, Stupid') implementation was to research the separation of concern between security related functionality and the

[1] See http://www.chu-charleroi.be/kmehr/htm/kmehr.htm for specification. This is a Belgian national standard, it is not used elsewhere.

[2] See for example the IHE standards concerning IHE XDS (document exchange).

'other' functionality of a software application (in view of the 'vertical' nature of security).

3.1.2 Choice as Integration Scenario

The scenario chosen for the integration trial reflects effectively the current needs in the healthcare domain. As explained in the previous version of D9.1, information exchange is a top priority, and the 'distributed health repository with central access' concept is currently being deployed all over Europe and the USA. The link with the previously elaborated, but more user-centred, scenario of Personal Health Records is clear. The information storage and retrieval flow is technically no different for result servers, summary records or even PHRs. In the end, the goal of increased electronic information exchange should also be to make as much of this HCP exchange as possible more user-centric (at least for disclosure control, cf. D9.1 first iteration and the Dutch National Switchpoint).

The selection of this integration scenario is also determined by the availability of 'in-house' developed application software which includes a full-blown security implementation for the 'bottom-up' approach to TAS^3 integration. The legacy application described already includes an elaborate functional security implementation. This means that from a technical architecture perspective, 'placeholders' for TAS^3 components are already present (for at least the majority of components); this is not the case with many legacy applications used in the field. Together with the fact that the architecture and code is well known, this allows WP9 to concentrate effort fully on the actual integration issues and the demonstration of TAS^3 components.

Finally, as explained later in this document, the original application (which has security built inside) is an excellent benchmark for measuring TAS^3 added value.

3.2 Application Description of Usage

3.2.1 Use Cases

The use scenario for a 'document repository' is information storage and retrieval. The PILS application is oriented towards professional use, but there is no reason why the access policies could not be changed to allow patients to look for their own data (in fact the generic security system encourages such an evolution). As an example for practical use, two scenarios can be kept in mind: use of the repositories as a hospital results server and as a summary record repository. A hospital results server would contain medical reports on episodes of hospital care. A summary server would contain 'summary records' created by primary care physicians. A summary record is, roughly speaking, a set of data that a physician needs in order to understand the medical status of the patient in just a few minutes, and to ensure continuity of care. The main difference between the two is that a results document does not usually need fine-grained access control, whereas a summary record does (although this is subject to discussion).

	Create/Update/Delete	Retrieve
Hospital results server	Hospital Information System	Health Care Professionals

		Currently Physicians Only
Summary record	GP – GMD Holder[3]	Health Care Professionals Currently Physicians Only

Eventually, abstracting from the security interactions (TAS³'s responsibility), the flow is basically the same in all information retrieval use cases:

- Physicians lack information about a patient. This information is not stored locally (even distributed)
- They use PILS as a central contact point to search for the desired information.

In this first integration trial, the use cases focus on information retrieval: information upload is not explicitly addressed. Repositories are considered to already contain data.

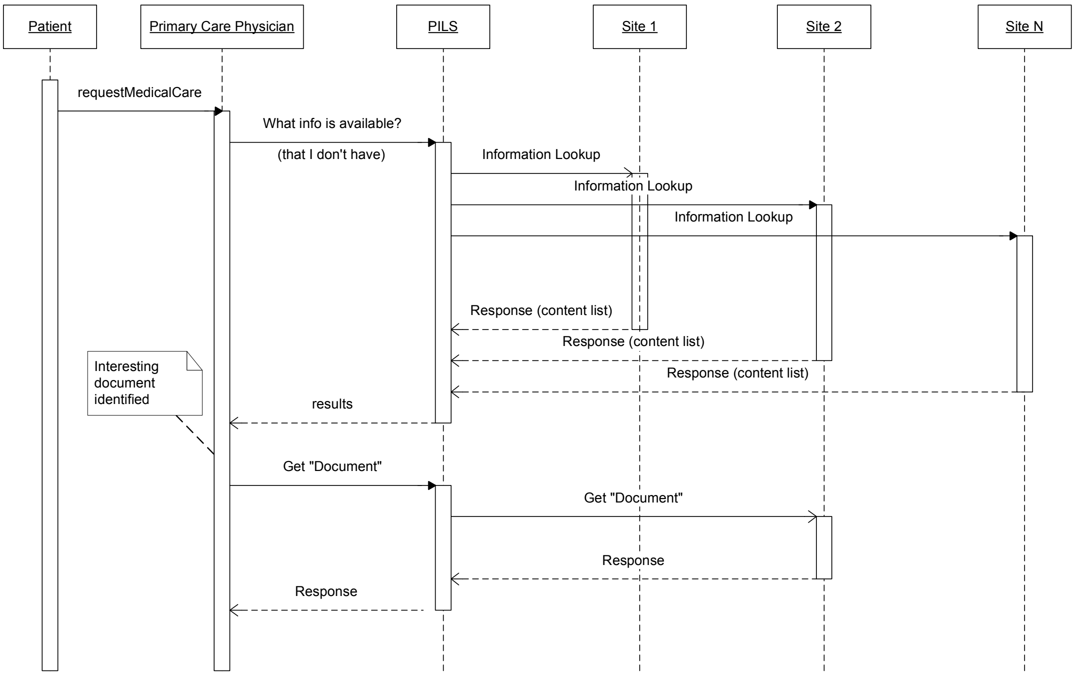
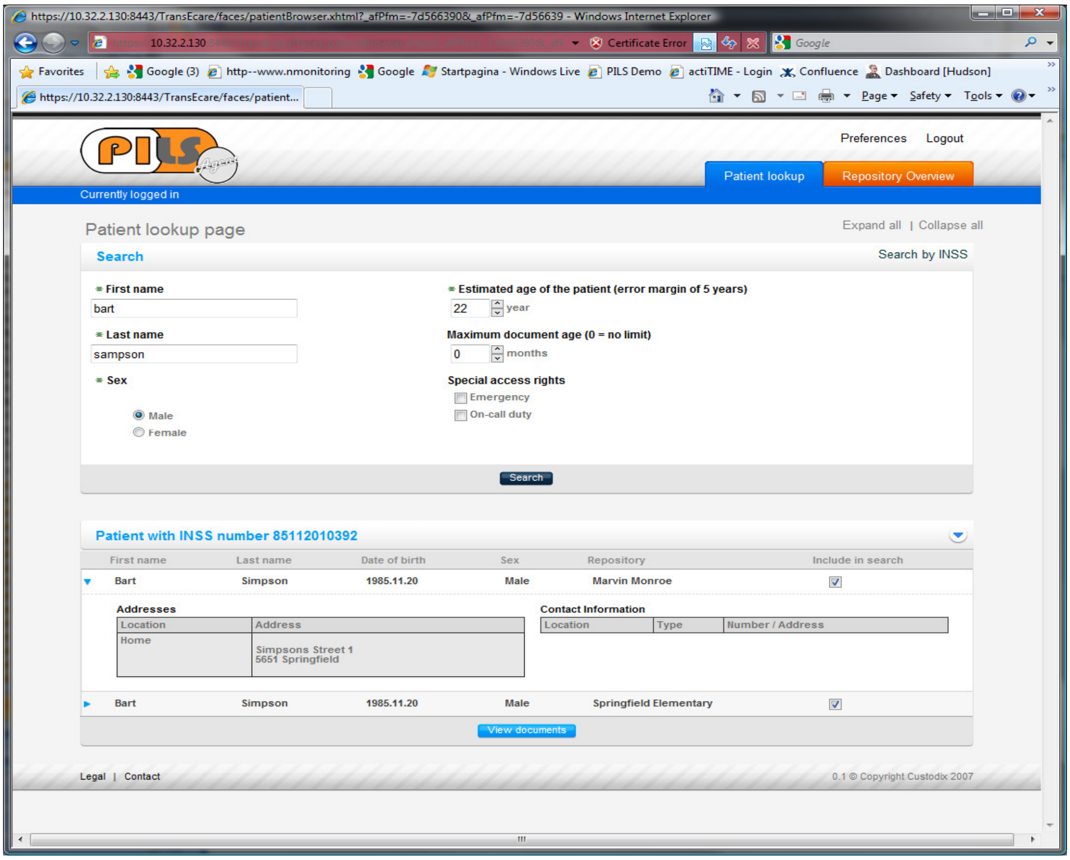


Figure 1: Information retrieval

[3] GMD Holder: Each patient can appoint a physician who then becomes the main person responsible for maintaining his or her medical history. It is the 'family' GP. GMD stands for "Globaal Medisch Dossier" or "Global Medical Record".

3.2.2 Application Description (Belgian context)

The starting point for describing PILS/repository operations is a physician who needs medical information on a specific patient (in this use case, one can assume that patients are uniquely identified through a social security number and physicians through a doctor registration number). The physician does not have this information stored locally, and does not even necessarily know if it is available or where it is stored. To start his search, he will contact the Patient Information Location Service (PILS). In order to login, he will need to authenticate using his eID (Belgian electronic Identity Card, which supports strong authentication). As the PILS access policy currently restricts use to physicians, the user will need to prove that he is indeed a doctor by obtaining a PILS-accepted 'physician credential' (see below). Once the user has access to the PILS, he can search for patients based on a number of identifiers[4] (Figure 2).



Patient lookup page

Search

* First name: bart

* Last name: sampson

* Sex: ● Male ○ Female

* Estimated age of the patient (error margin of 5 years): 22 year

Maximum document age (0 = no limit): 0 months

Special access rights: □ Emergency □ On-call duty

Patient with INSS number 85112010392

First name	Last name	Date of birth	Sex	Repository	Include in search
Bart	Simpson	1985.11.20	Male	Marvin Monroe	☑
Bart	Simpson	1985.11.20	Male	Springfield Elementary	☑

Addresses

Location	Address
Home	Simpsons Street 1 5651 Springfield

Contact Information

Location	Type	Number / Address

Figure 2: PILS patient query screen with results shown (lower section)

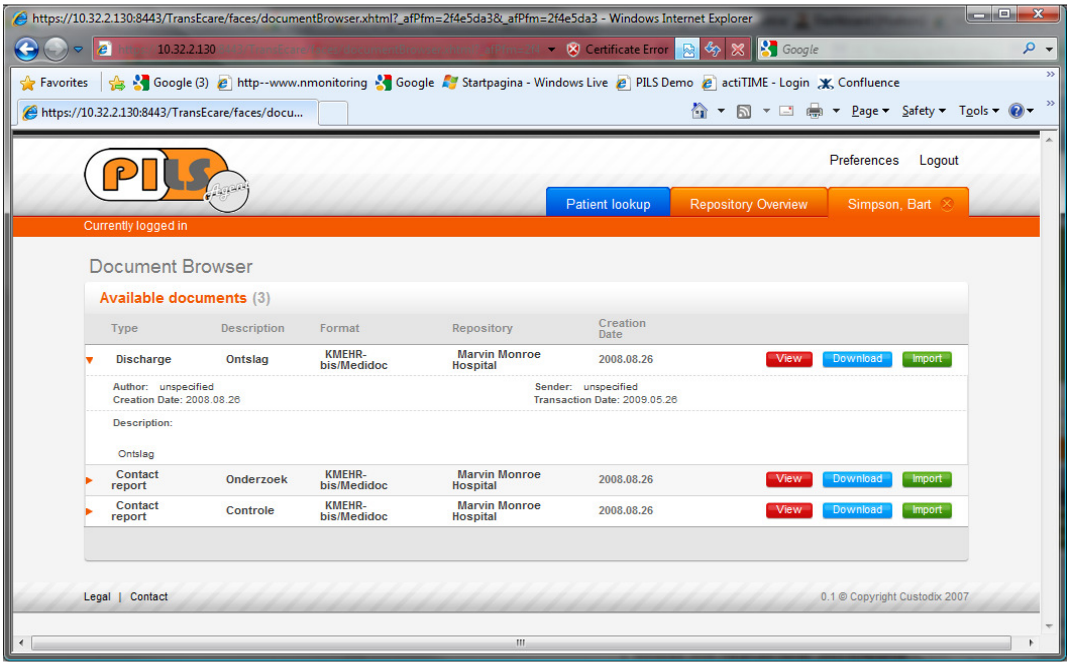
[4] An HCP needs to provide sufficiently detailed information about a patient before search is allowed. Furthermore, if too many different patients (>5) correspond with the query, no results are shown and the user first has to refine his query. These precautions are made to prevent abuse. One the one hand, Physicians are only allowed to search for information about patients that they treat. On the other hand, there is no way to know up-front who is treating (or is going to treat) a patient as there is free choice of medical care in Belgium. See also further "therapeutic relationship".

The PILS will forward any query to the connected (registered) data sources[5]. Together with the query, the PILS forwards the user credentials that the user presented at logon. This allows sources to decide autonomously whether or not they will give access to a particular HCP, i.e. the local security policy has priority. Confidentiality is therefore guarded by the legitimate data controllers (disclosing service providers).

Typically, access to medical data requires proof of a relevant therapeutic relationship between physician and patient. Currently, no central repository for therapeutic relationships is available. Hence the only location where such information is available is often in the data source itself, which is a more practical reason why access decisions are managed locally.

The result of the query is a list of possible matching patients (and data repositories that contain data about those patients, see Figure 2), allowing the physician to select the data sources he wants to get detailed information from. Note that all returned information is covered by the access control policies. Hence, if a user has no access to any of the documents on a data resource (even though there are such documents), the data source will not show in the query[6].

After selecting the patient and the data sources from which data is desired, a list of available documents with a short description (medical metadata) accessible on distributed repositories is shown (see Figure 3).


https://10.32.2.130:8443/TransEcare/faces/documentBrowser.xhtml?_afPfm=2f4e5da3&_afPfm=2f4e5da3 - Windows Internet Explorer
Certificate Error
Google
Favorites
Google (3)
http--www.nmonitoring
Google
Startpagina - Windows Live
PILS Demo
actiTIME - Login
Confluence
https://10.32.2.130:8443/TransEcare/faces/docu...
Page
Safety
Tools
Preferences
Logout
Patient lookup
Repository Overview
Simpson, Bart
Currently logged in
Legal | Contact
0.1 © Copyright Custodix 2007

Document Browser

Available documents (3)

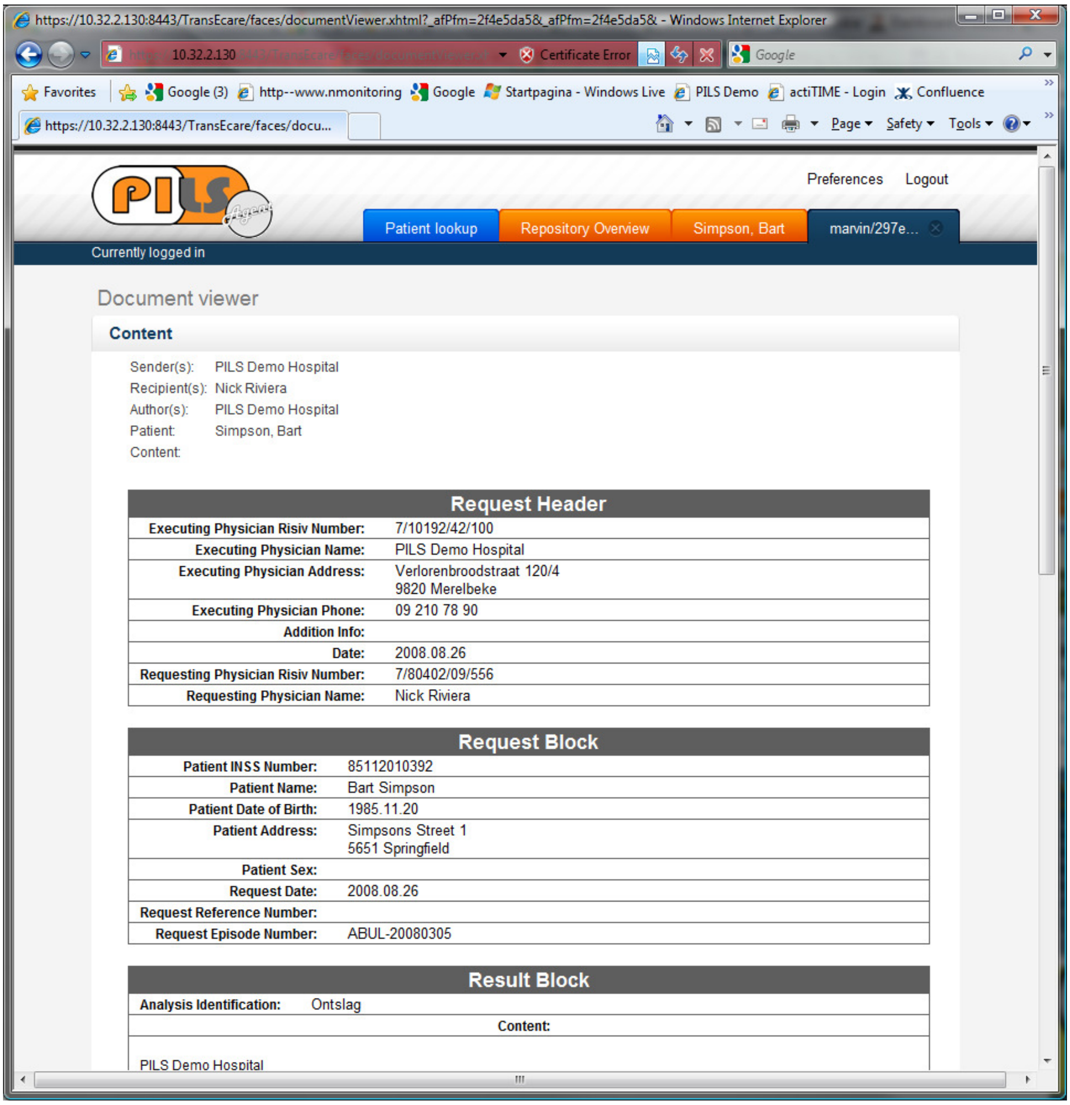
Type	Description	Format	Repository	Creation Date			
▼ Discharge	Ontslag	KMEHR-bis/Medidoc	Marvin Monroe Hospital	2008.08.26	View	Download	Import
Author: unspecified Creation Date: 2008.08.26			Sender: unspecified Transaction Date: 2009.05.26				
Description: Ontslag							
▶ Contact report	Onderzoek	KMEHR-bis/Medidoc	Marvin Monroe Hospital	2008.08.26	View	Download	Import
▶ Contact report	Controle	KMEHR-bis/Medidoc	Marvin Monroe Hospital	2008.08.26	View	Download	Import

Figure 3: Document query result

[5] Repositories must be preregistered to the PILS. Authentication is performed through SSL (X.509 based mutual authentication). TAS³ could provide a dynamic trust mechanism which would be able to replace this implementation.

[6] This is the desired behaviour over which consensus was reached when discussing with physicians, legal and ethical experts.

Subsequently, the appropriate information can be viewed or downloaded (see Figure 4).



Document viewer

Content

Sender(s): PILS Demo Hospital
Recipient(s): Nick Riviera
Author(s): PILS Demo Hospital
Patient: Simpson, Bart
Content:

Request Header	
Executing Physician Risiv Number:	7/10192/42/100
Executing Physician Name:	PILS Demo Hospital
Executing Physician Address:	Verlorenbroodstraat 120/4 9820 Merelbeke
Executing Physician Phone:	09 210 78 90
Addition Info:	
Date:	2008.08.26
Requesting Physician Risiv Number:	7/80402/09/556
Requesting Physician Name:	Nick Riviera

Request Block	
Patient INSS Number:	85112010392
Patient Name:	Bart Simpson
Patient Date of Birth:	1985.11.20
Patient Address:	Simpsons Street 1 5651 Springfield
Patient Sex:	
Request Date:	2008.08.26
Request Reference Number:	
Request Episode Number:	ABUL-20080305

Result Block	
Analysis Identification:	Ontslag
Content:	
PILS Demo Hospital	

Figure 4: KMEHR document view

3.3 Integration Trial Objective

The application described already includes an elaborate functional security implementation. The objective of this integration trial is gradually to replace existing security components with TAS³- compliant components. This will enable evaluation of the integration and interoperability of the TAS3 components on a technical level.

The aim is gradually (as they become available) to integrate the following TAS³ components:

- Identity Provision
- Authentication
- Trusted communication

- Logging
- Policy Decision
- Logging
- Policy Control (Dashboard)

This bottom-up approach (i.e. starting from a working legacy application) has the advantage that it can provide TAS³ with direct feedback about practical (technical) difficulties and requirements; and that the TAS³-ified end goal can be evaluated using the original application as a benchmark. The aim is to:

- establish that TAS³ can remove the burden of implementing complex security such as that implemented in the legacy application (i.e. usability towards developers).
- establish that TAS³ offers a wider set of security functionality than commonly encountered, focusing especially on putting the data subject in control (i.e. in this case leading to patient empowerment).

3.4 Legacy Architecture

3.4.1 Identity Provider

The functionality of the PILS portal web application as described above includes authentication through eID and users obtaining credentials to prove their accreditation as a physician. The implementation with the SAML Identity Provider (IdP) is explained below.

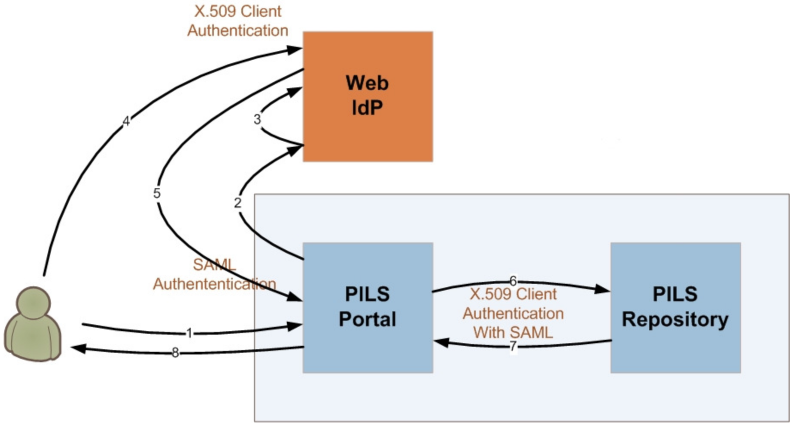

X.509 Client Authentication
Web IdP
4
3
5
2
SAML Authententication
PILS Portal
6
X.509 Client Authentication With SAML
7
PILS Repository
1
8

Figure 5: PILS Legacy Authentication and Credential Provision

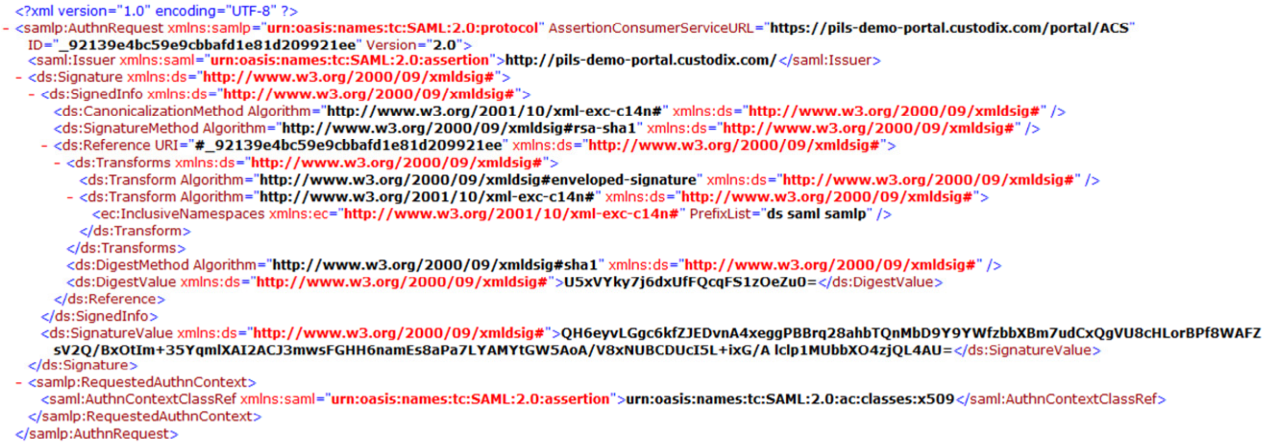
Figure 5 illustrates the authentication and credential provisioning:

1. The user visits the PILS portal and chooses an authentication method (The actual implementation is restricted to eID authentication).
2. The PILS portal creates a SAML request targeted at the IdP containing the authentication method chosen by the user (i.e. the eID IDP).

3. At the IdP, a servlet intercepts this request and inspects it to see which authentication method is requested. The user is being redirected to the page corresponding with the authentication method.
4. The user provides his credentials and logs in using the chosen authentication method. eID authentication happens through the standard X.509 HTTPS profile which is supported by all browsers (with private key on the smartcard).
5. The IdP uses the Security Context set up during authentication to handle the SAML Request and creates a SAML Response. This response is pushed to the Portal through a HTTP Post.
    - The SAML response includes an authentication assertion and a custom attribute assertion which lists the unique HCP identifier (RIZIV code) for the user. The link between identity (i.e. eID and HCP identifier is currently stored in the IdP itself. The Belgian government has planned to provide such a credential service for some time, but it currently seems restricted to internal use.
6. The Portal fetches the SAML assertion and authenticates the user based on the SAML Assertion. When the user then performs an action which requires communication with the Repository, the Portal adds the SAML assertion of the user to the Web Service call according to the WS-Security specification. The communication between the Portal and Repository is based on SSL (mutual authentication required).
    - The SAML credential provided by the Identity Provider is targeted towards the PILS application: it is forwarded verbatim to the different repositories. Those repositories accept this assertion (even though it is not targeted at them) because there is an established (by contract) static trust relation between PILS and the repositories. Strictly speaking (according to the standard), credentials should be re-issued for each of the repositories, or the target should be the PILS-repository 'trust-group' (e.g. a URN).
7. The Repository sends a response to the PILS Portal.
8. The Portal shows the result to the user.

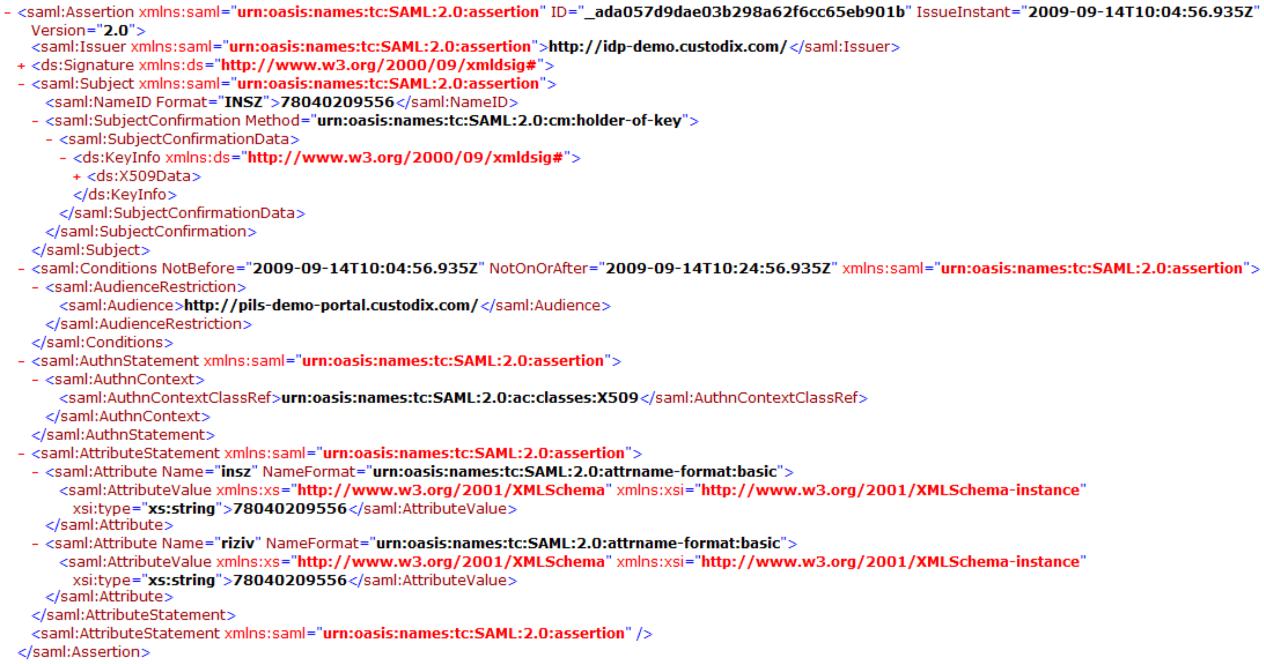
Technology used:

1. Java Platform
2. OpenSAML 2 library
3. Backend: Spring 2 Framework / Hibernate ORM / MySQL DB
4. Web Server/Servlet Container: Tomcat



```
  <?xml version="1.0" encoding="UTF-8" ?>
- <samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" AssertionConsumerServiceURL="https://pils-demo-portal.custodix.com/portal/ACS"
    ID="_92139e4bc59e9cbbafd1e81d209921ee" Version="2.0">
    <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">http://pils-demo-portal.custodix.com/</saml:Issuer>
  - <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    - <ds:SignedInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" xmlns:ds="http://www.w3.org/2000/09/xmldsig#" />
        <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" xmlns:ds="http://www.w3.org/2000/09/xmldsig#" />
      - <ds:Reference URI="#_92139e4bc59e9cbbafd1e81d209921ee" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        - <ds:Transforms xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
            <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" xmlns:ds="http://www.w3.org/2000/09/xmldsig#" />
          - <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
              <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="ds saml samlp" />
            </ds:Transform>
          </ds:Transforms>
          <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" xmlns:ds="http://www.w3.org/2000/09/xmldsig#" />
          <ds:DigestValue xmlns:ds="http://www.w3.org/2000/09/xmldsig#">U5xVYky7j6dxUfFQcqFS1zOeZu0=</ds:DigestValue>
        </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue xmlns:ds="http://www.w3.org/2000/09/xmldsig#">QH6eyvLGgc6kfZJEDvnA4xeggPBBrq28ahbTQnMbD9Y9YWfzbbXBm7udCxQgVU8cHLorBPf8WAFZ
        sV2Q/BxOtIm+35YqmlXAI2ACJ3mwsFGHH6namEs8aPa7LYAMYtGW5AoA/V8xNUBCDUcI5L+ixG/A lclp1MUbbXO4zjQL4AU=</ds:SignatureValue>
    </ds:Signature>
  - <samlp:RequestedAuthnContext>
      <saml:AuthnContextClassRef xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">urn:oasis:names:tc:SAML:2.0:ac:classes:x509</saml:AuthnContextClassRef>
    </samlp:RequestedAuthnContext>
  </samlp:AuthnRequest>
```

Listing 1: SAMLrequest



```
- <saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="_ada057d9dae03b298a62f6cc65eb901b" IssueInstant="2009-09-14T10:04:56.935Z"
    Version="2.0">
    <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">http://idp-demo.custodix.com/</saml:Issuer>
  + <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  - <saml:Subject xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
      <saml:NameID Format="INSZ">78040209556</saml:NameID>
    - <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
      - <saml:SubjectConfirmationData>
        - <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
          + <ds:X509Data>
          </ds:KeyInfo>
        </saml:SubjectConfirmationData>
      </saml:SubjectConfirmation>
    </saml:Subject>
  - <saml:Conditions NotBefore="2009-09-14T10:04:56.935Z" NotOnOrAfter="2009-09-14T10:24:56.935Z" xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
    - <saml:AudienceRestriction>
        <saml:Audience>http://pils-demo-portal.custodix.com/</saml:Audience>
      </saml:AudienceRestriction>
    </saml:Conditions>
  - <saml:AuthnStatement xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
    - <saml:AuthnContext>
        <saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:X509</saml:AuthnContextClassRef>
      </saml:AuthnContext>
    </saml:AuthnStatement>
  - <saml:AttributeStatement xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
    - <saml:Attribute Name="insz" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
        <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:type="xs:string">78040209556</saml:AttributeValue>
      </saml:Attribute>
    - <saml:Attribute Name="riziv" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
        <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:type="xs:string">78040209556</saml:AttributeValue>
      </saml:Attribute>
    </saml:AttributeStatement>
    <saml:AttributeStatement xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" />
  </saml:Assertion>
```

Listing 2: SAML assertion

3.4.2 PILS Portal

The application is described above.

Technology used:

1. Java Platform
2. Frontend: JSF & WS through Jax-WS
3. Backend: Spring 2 Framework
4. Web Server/Servlet Container: Tomcat

3.4.3 Repository

3.4.3.1 Interface

The repository is a straightforward document-based store and retrieve repository. It stores KMEHR documents. Uploaded documents are validated (for formatting and consistency) and a limited set of metadata is extracted for indexing (search functionality). This simple implementation does not support document requiring fine-grained access control (e.g. on separate sections in the document) in its given form. This functionality will be added as necessary for the further demonstrations and pilots (the generic access control mechanisms are sufficiently flexible for this, see below).

The repository supports the following (self-explanatory) services:

- DeleteDocumentService
    - Delete a document by unique document ID.
- DownloadDocumentService
    - Download a document by unique document ID.
- ListDocumentsService
    - List documents related to a patient ID.
- ListPatientsService
    - List patient information
- UploadDocumentService
    - Upload a document

3.4.3.2 Technology

Technology Used:

1. Java Platform
2. Frontend: WS using Jax-WS
3. Backend: Spring 2 Framework / Hibernate ORM / MySQL DB
4. Web Server/Servlet Container: Tomcat
5. Security interceptors: based on Acegi Security Framework (now Spring Security)
6. XACML library/PDP: Modified SunXACML library

3.4.3.3 Access Control Architecture

The repository contains patient documents inserted by physicians. The document format is restricted to KMEHR only. The front-end of the repository is a Web Services (WS) interface, each service method is mapped to a Web Service invocation. The repository uses XACML (eXtensible Access Control Markup Language) based access control to post-filter service method result sets. When access control rules change, it should suffice to modify the appropriate XACML policy, no code changes are necessary.

The architecture of the repository aims to separate the security functionality from the rest of the application (application changes should no longer affect the security implementation, changes in access rules should not require implementation changes). The functionality resides in the Service Layer and beyond (Business and DAO layer). The security mechanics are triggered at the Service Layer by intercepting Service Layer and DAO layer method invocations, and forwarding an authorization question to an independent XACML PDP.

Figure 6 shows the architecture and control flow of the repository. The repository can be accessed using a Web Service front end. When a service call is made, a Security Context is set up by a ServletFilter. The information about the principal (i.e. the user who is logged into the PILS) making the Web Service call is passed in the WS call security metadata by the portal to the callee. Currently it is necessary that trust is established *a-priori* between the Repository client and the Repository itself (e.g. trust is established through SSL with mutual authentication). The attributes in the Security Context are used later on when information about the principal is needed by the Context Handler.

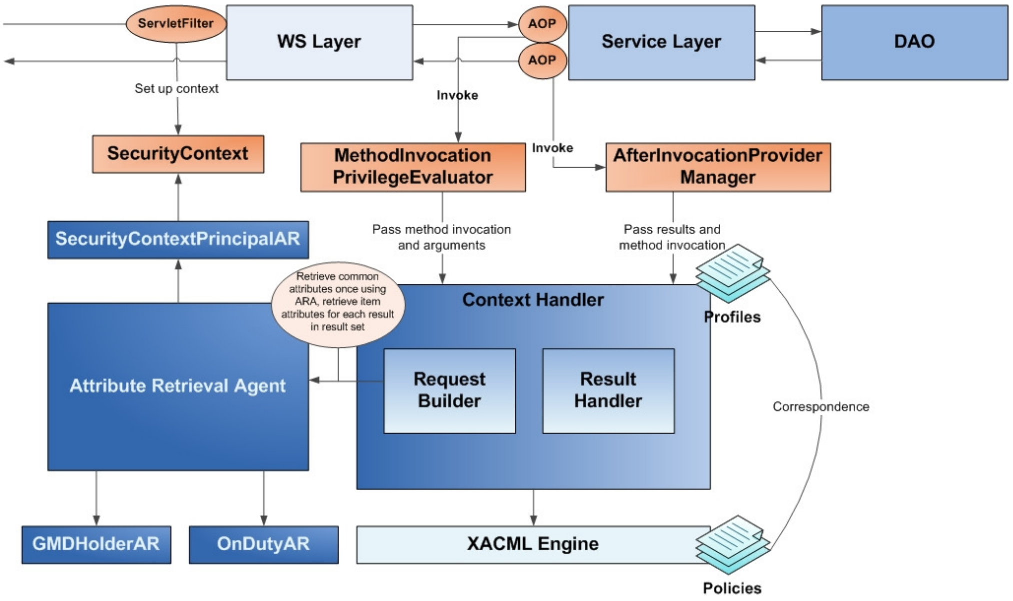

ServletFilter
WS Layer
AOP
AOP
Service Layer
DAO
Set up context
Invoke
Invoke
SecurityContext
MethodInvocation PrivilegeEvaluator
AfterInvocationProvider Manager
SecurityContextPrincipalAR
Pass method invocation and arguments
Pass results and method invocation
Retrieve common attributes once using ARA, retrieve item attributes for each result in result set
Context Handler
Profiles
Attribute Retrieval Agent
Request Builder
Result Handler
Correspondence
GMDHolderAR
OnDutyAR
XACML Engine
Policies

Figure 6: Repository Architecture

The actual access control is performed on the method invocations at the Service layer (or DAO). These methods are intercepted according to the AOP (Aspect Oriented Programming) paradigm and the method, its arguments and its return value (only in case of post-invocation interception) are passed to the Context Handler. The AOP interceptors can be considered to be PEPs.

One of the difficult issues is bridging the gap between application-specific method arguments and return values and a generic description of subjects and objects in an Access Control policy. For this, a mapping from application-specific constructs to conceptual 'enterprise-level' constructs (thus allowing reuse and uniform enforcement of the same access control policies over the enterprise) is defined in 'profiles'.

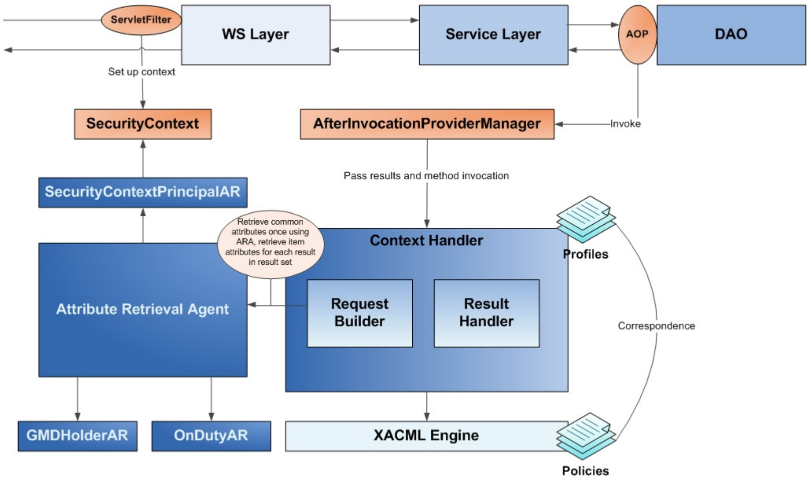
A generic Context Handler uses method-specific profiles (reflected in configuration files) which describe the way XACML requests should be constructed. They indicate which attributes are required to fulfil the request (the implementation requires all attributes to be pushed *a-priori*[7]).

The Context Handler uses the method signature and the access control type (either before or after invocation) to determine which profile should be used. If no matching profile is found, the method execution is not allowed (before invocation) or the method result is emptied (after invocation), i.e. a 'deny all' policy is enforced.

The profile contains a list of required attributes and how these attributes can be resolved. Attributes are obtained from either the method call, the Security Context or another independent attribute provider. Once all required attributes are retrieved, the Request Builder forwards the XACML request to the XACML Engine which then uses its policies to make a decision.

In case of post-invocation access control, the Result Handler uses this decision to filter the result set and removes all objects for which no access was granted by the XACML engine (again this is a generic implementation which relies on the profiles for the application specific translation). In case of pre-invocation access control, the Context Handler returns a Boolean indicating whether the method invocation is allowed or not.

Note that policy enforcement can also be invoked on the DAO layer as depicted below (even in the same application):


ServletFilter
WS Layer
Service Layer
AOP
DAO
Set up context
SecurityContext
AfterInvocationProviderManager
Invoke
Pass results and method invocation
SecurityContextPrincipalAR
Retrieve common attributes once using ARA, retrieve item attributes for each result in result set
Context Handler
Profiles
Attribute Retrieval Agent
Request Builder
Result Handler
Correspondence
GMDHolderAR
OnDutyAR
XACML Engine
Policies

[7] Dynamic attribute retrieval following an "missing attribute" PDP reply is not implemented.

Figure 7: DAO layer interceptors

3.4.3.4 Attribute Retrieval Profiles

Attribute Retrieval Profiles indicate which attributes need to be retrieved for a Service Layer or DAO method.

The example below shows the Profile is will be used to resolve attributes for the ‘find’ method on the PatientService after invocation. The ‘find’ DAO method is a patient look-up method, which returns patient objects.

The example illustrates the profile for a generic after invocation interceptor. This interceptor interprets the return value as a collection of XACML resources, issues XACML requests for each of them and filters out those resources to which access has been denied by the XACML PDP. Thus, XACML ‘DENY’ responses never result in a failed call with an exception being thrown, but return values are filtered out (even if there is only one return value)[8].

The interceptor knows two kind of attributes: common and item attributes. Common attributes are retrieved once and are the same for each resource request. Item attributes are retrieved for each item in the returned collection and are thus different for each resource request.

```
<AttributeRetrievalProfile
xmlns="http://www.custodix.com/xacml/profiles"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <Target>
   <ServiceMethod
accessControlType="afterInvocation">com.custodix.scare.dao.PatientS
ervice.find</ServiceMethod>
 </Target>
 <RequiredAttributes>

   <CommonAttributes>
     <SubjectAttribute
       id="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
       type="http://www.w3.org/2001/XMLSchema#string">
       <Retriever source="SecurityContextPrincipalAR"
         attribute="id" />
     </SubjectAttribute>
     <SubjectAttribute
id="urn:custodix:healthcare:0.1:attribute:hcp-type"
       type="http://www.w3.org/2001/XMLSchema#string">
       <Retriever source="SecurityContextPrincipalAR"
         attribute="hcType" />
     </SubjectAttribute>
     <SubjectAttribute id="urn:custodix:healthcare:0.1:attribute:on-call-
duty"
       type="http://www.w3.org/2001/XMLSchema#boolean">
       <Retriever source="OnCallDutyAR"
         attribute="onCallDuty" >

 <RetrievalArgument>urn:oasis:names:tc:xacml:1.0:subject:subject-
id</RetrievalArgument>
```

[8] This implies that this kind of interception should not be used for methods with possible side effects (since there will be no transaction rollback)

```
          </Retriever>
        </SubjectAttribute>
        <EnvironmentAttribute
  id="urn:custodix:healthcare:0.1:environment:call-duty-ack"
          type="http://www.w3.org/2001/XMLSchema#boolean">
          <Retriever source="SecurityContextPrincipalAR"
            attribute="callDutyAck" />
        </EnvironmentAttribute>
        <EnvironmentAttribute
          id="urn:oasis:names:tc:xacml:1.0:environment:current-time"
          type="http://www.w3.org/2001/XMLSchema#dateTime">
          <CurrentTime />
        </EnvironmentAttribute>
        <ResourceAttribute
          id="urn:custodix:application:kmehrrepository:0.1:resource-type"
          type="http://www.w3.org/2001/XMLSchema#string"
          retrievalMethod="getType"/>
      </CommonAttributes>

      <ItemAttributes>
        <ResourceAttribute id="urn:custodix:healthcare:0.1:patient-id"
          type="http://www.w3.org/2001/XMLSchema#string"
          retrievalMethod="getId" />
        <SubjectAttribute id="urn:custodix:healthcare:0.1:attribute:hcp-
  relation:gmd-holder-of"
          type="http://www.w3.org/2001/XMLSchema#string">
          <Retriever source="GMDHolderAR" attribute="gmdHolderOf">
            <RetrievalArgument>urn:custodix:healthcare:0.1:patient-
  id</RetrievalArgument>
          </Retriever>
        </SubjectAttribute>
      </ItemAttributes>

    </RequiredAttributes>
  </AttributeRetrievalProfile>
```

Listing 3:Attribute Retrieval Profile

Consider the item attributes of this example:

```
  <ResourceAttribute id="urn:custodix:healthcare:0.1:patient-id"
  type=http://www.w3.org/2001/XMLSchema#string retrievalMethod="getId" />
```

For each of the items (which are patient objects) in the returned collection, the XACML access request will be annotated with the resource attribute 'urn:custodix:healthcare:0.1:patient-id' (which is a string), which will be retrieved by invoking the `getId` method on that patient object. In other words, the patient ID of the patient will be included in the access request. This mechanism makes it possible for access control policies to include rules which depend on the attributes of the protected object themselves (e.g. this particular patient, people under 18, etc.)

```
  <SubjectAttribute id="urn:custodix:healthcare:0.1:attribute:hcp-relation:gmd-
  holder-of"
                             type="http://www.w3.org/2001/XMLSchema#string">
   <Retriever source="GMDHolderAR" attribute="gmdHolderOf">
     <RetrievalArgument>urn:custodix:healthcare:0.1:patient-
  id</RetrievalArgument>
  </Retriever>
</SubjectAttribute>
```

Equally, the above configuration indicates that a subject attribute ‘urn:custodix:healthcare:0.1:attribute:hcp-relation:gmd-holder-of’[9] will be included in each resource request. The attribute can be retrieved by querying the GMD holder Attribute Resolver (GMDHolderAR) for the GMD Holder of the patient (urn:custodix:healthcare:0.1:patient-id)[10].

3.4.3.5 Example Policy

Listing 4 shows an example XACML policy for reading KMEHR documents from the repository. The rule stipulates that documents may be read if the physician is either the recipient of the document or the GMD-Holder for the patient in the document. Listing 5 shows an extract from the matching profile which help clarify the XACML rules.

```
<!-- ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++  -->
<!-- PERMIT-OVERRIDES
<!-- ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++  -->
<Policy xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xacmlcontext="urn:oasis:names:tc:xacml:2.0:context:schema:os"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-
schema-os.xsd"
PolicyId="urn:oasis:names:tc:xacml:2.0:example:policyid:2"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
algorithm:permit-overrides">

 <VariableDefinition VariableId="CheckIfGMDHolder">
   <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
is-in">
      <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-
only">
        <ResourceAttributeDesignator
AttributeId="urn:custodix:healthcare:0.1:patient-id"
DataType="http://www.w3.org/2001/XMLSchema#string" />
      </Apply>
      <SubjectAttributeDesignator
AttributeId="urn:custodix:healthcare:0.1:attribute:hcp-
relation:gmd-holder-of"
DataType="http://www.w3.org/2001/XMLSchema#string" />
   </Apply>
 </VariableDefinition>

 <!-- +++++++++++++++++++++++++++++++++++++++++++++++++++++++++  -->
 <!-- This policy only applies to reading "kmehrdocument" resources
```

[9] This refers to a specific therapeutic relationship between patient and physician in Belgium. Each patient can have a ‘GMD holder’ which is the primary care physician who is responsible for maintaining a record with all medical events concerning that patient.

[10] This requires a GMD Holder Attribute Resolver to be defined which implements a method ‘gmdHolderOf” which can deal with a patient identifier as argument. Where this AR retrieves it information is not important, this can be a local database or an external identity provider.

```
 -->
  <!-- ++++++++++++++++++++++++++++++++++++++++++++++++++++++++  -->
  <Target>
    <Resources>
      <Resource>
        <ResourceMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">documententry</A
ttributeValue>
          <ResourceAttributeDesignator
AttributeId="urn:custodix:application:kmehrrepository:0.1:resource-
type"

DataType="http://www.w3.org/2001/XMLSchema#string" />
        </ResourceMatch>
      </Resource>
    </Resources>
    <Actions>
      <Action>
        <ActionMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">read</AttributeV
alue>
          <ActionAttributeDesignator
AttributeId="urn:custodix:healthcare:0.1:action"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </ActionMatch>
      </Action>
    </Actions>
  </Target>

  <!-- +++++++++++++++++++++++++++++++++++++++++++++++++++++++++  -->
  <Rule RuleId="DenyByDefault" Effect="Deny">
    <Description> Standard rule.. default deny all (policy combiner permit-
    overrides)</Description>
  </Rule>

  <!-- +++++++++++++++++++++++++++++++++++++++++++++++++++++++++  -->
  <Rule RuleId="GMDHolderOf" Effect="Permit">
    <Description>
    GMD holders may see their patients documents.
    Note this is the same rule as for "patiententry" resources and
should be put
    in a "global" policy.
    </Description>
    <Condition>
      <VariableReference VariableId="CheckIfGMDHolder" />
    </Condition>
  </Rule>

  <!-- ++++++++++++++++++++++++++++++++++++++++++++++++++++++++  -->
  <Rule RuleId="recipients" Effect="Permit">
    <Description>
    Recipients logically may see messages sent to them
    </Description>
    <Condition>
      <Apply
```

```
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-is-in">
       <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-
only">
         <SubjectAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"

DataType="http://www.w3.org/2001/XMLSchema#string" />
       </Apply>
       <ResourceAttributeDesignator
AttributeId="urn:custodix:healthcare:0.1:kmehr:recipients"

DataType="http://www.w3.org/2001/XMLSchema#string" />
     </Apply>
   </Condition>
 </Rule>
</Policy>
```

Listing 4: Example Policy accessing a document

```
<AttributeRetrievalProfile xmlns="http://www.custodix.com/xacml/profiles"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <Target>
   <ServiceMethods>
     <ServiceMethod
accessControlType="beforeInvocation">com.custodix.pils.repository.services.D
ownloadDocumentService.getDocumentData</ServiceMethod>
   </ServiceMethods>
 </Target>

 <RequiredAttributes>
   <CommonAttributes>
     <SubjectAttribute
       id="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
       type="http://www.w3.org/2001/XMLSchema#string">
       <Retriever source="SecurityContextPrincipalAR"
         attribute="id" />
...

     <EnvironmentAttribute
       id="urn:oasis:names:tc:xacml:1.0:environment:current-time"
       type="http://www.w3.org/2001/XMLSchema#dateTime">
       <CurrentTime />
     </EnvironmentAttribute>

     <ResourceAttribute
       id="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
       type="http://www.w3.org/2001/XMLSchema#string">
       <Retriever source="ServiceMethodArgumentAR" attribute="0" />
     </ResourceAttribute>

     <ResourceAttribute
       id="urn:custodix:application:kmehrrepository:0.1:resource-type"
       type="http://www.w3.org/2001/XMLSchema#string">
       <Retriever source="DocumentResourceAR" attribute="getResourceType">
         <RetrievalArgument>urn:oasis:names:tc:xacml:1.0:resource:resource-
id</RetrievalArgument>
       </Retriever>
     </ResourceAttribute>
```

```
      <ResourceAttribute id="urn:custodix:healthcare:0.1:kmehr:recipients"
        type="http://www.w3.org/2001/XMLSchema#string">
        <Retriever source="DocumentResourceAR" attribute="getRecipientIds">
          <RetrievalArgument>urn:oasis:names:tc:xacml:1.0:resource:resource-
id</RetrievalArgument>
        </Retriever>
      </ResourceAttribute>

...

      <ActionAttribute
        id="urn:custodix:healthcare:0.1:action"
        type="http://www.w3.org/2001/XMLSchema#string">
        <StringValue>read</StringValue>
      </ActionAttribute>
   </CommonAttributes>
 </RequiredAttributes>
</AttributeRetrievalProfile><?xml version="1.0" encoding="UTF-8"?>
```

Listing 5: Document Read Profile

3.4.3.6 Access Control and Granularity

In the current implementation, the smallest unit for access control is a KMEHR document. As earlier mentioned, this suits the requirements of medical outcomes quite well, but (according to some[11]) summary record access requires a finer-grained access control (based the sections in the document). The generic (application) security architecture allows for this change to be made with limited effort, so this will be done when needed to demonstrate the full range of TAS^3 capabilities.

3.5 Integration Plan

3.5.1 Introduction

The integration of the TAS^3 components is a phased process. First, initial (testable) versions of components become available (or reach a stable stage) Gradually. Secondly, component development is itself an iterative process in which functionality for the more complex TAS^3 processes will only be added gradually during the course of the project. Finally, the integration trials are intended to provide feedback and debugging to the component developers. This step-by-step approach to integration is therefore a natural process, arising partially from component dependency (i.e. dependency on a bugged component can block integration until the component is fixed). Each iteration (phase) of this process consists of four steps:

[11] Opinions amongst physicians, policy makers and patients tend to differ strongly on this point.

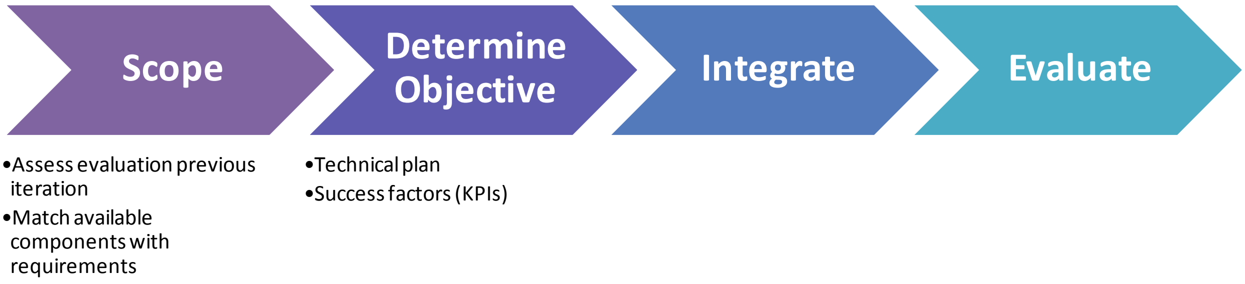

Scope
Determine Objective
Integrate
Evaluate
•Assess evaluation previous iteration
•Match available components with requirements
•Technical plan
•Success factors (KPIs)

Figure 8: Integration steps for each phase

The legacy PILS/Repository application used for the integration trial consists of four independent services deployed in a test setup at the Custodix Data Centre (Figure 9):

- PILS portal (will be accessible at http://tas3-portal.custodix.com/)[12]
- Identity Provider Service
- Repository Service 1 (will be deployed at 'tas3-repo2.custodix.com'), filled with dummy data
- Repository Service 2 (will be deployed at 'tas3-repo2.custodix.com'), filled with dummy data

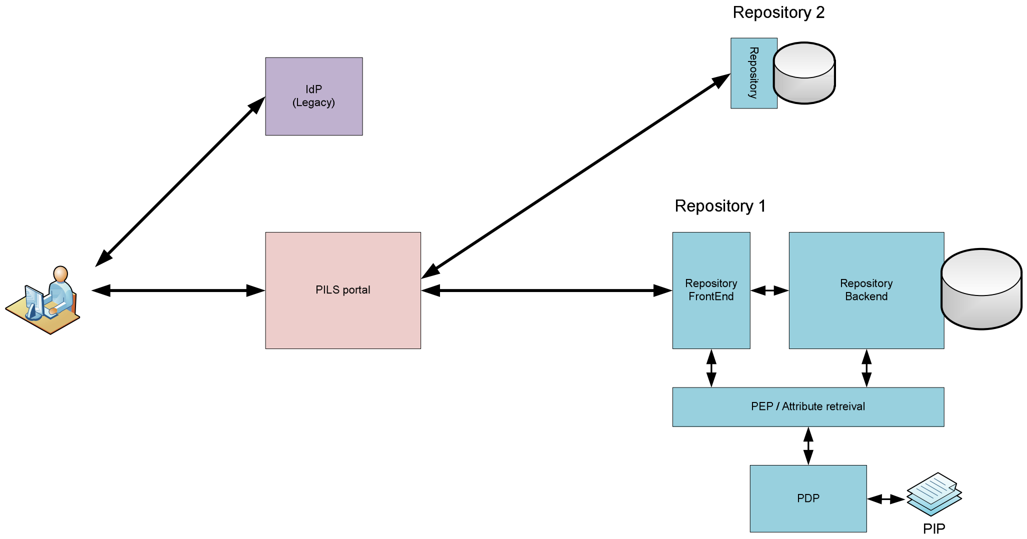


Figure 9: Integration trial services

3.5.2 Phase 1

The first integration phase aims to replace basic existent security functionality with TAS³ compliant implementations. The focus is on reaching compliance with

[12] Services are only guaranteed to be available during the review

the TAS³ protocol stack, rather than on integrating the extended TAS³ security functionality (for which implementations are not yet available) into the application workflow.

Scope

Note that most TAS³ components are only available in alpha versions with limited functionality at this stage in the project, and still have some instability issues.

	Component	Notes (status 11/2009)
PILS/Repository legacy implementation	Portal application (PILS)	Legacy, Functional
	Repository	Legacy, Functional
	Legacy IdP (SAML compliant)	Legacy, Functional
TAS3-ZXID component suite	TAS³-ZXID IdP	The ZXID components are at the time of writing not yet interoperating with the TAS³ auditing system.
	TAS³-ZXID Discovery Service	Under development
	TAS³-ZXID SSO SP endpoint (Java)	Depends on discovery service
	TAS³-ZXID SP communication library	Depends on discovery service Exposed API not yet available
TAS³-Logging components	TAS³ Audit Bus	Basic TAS³ components do not yet integrate this.

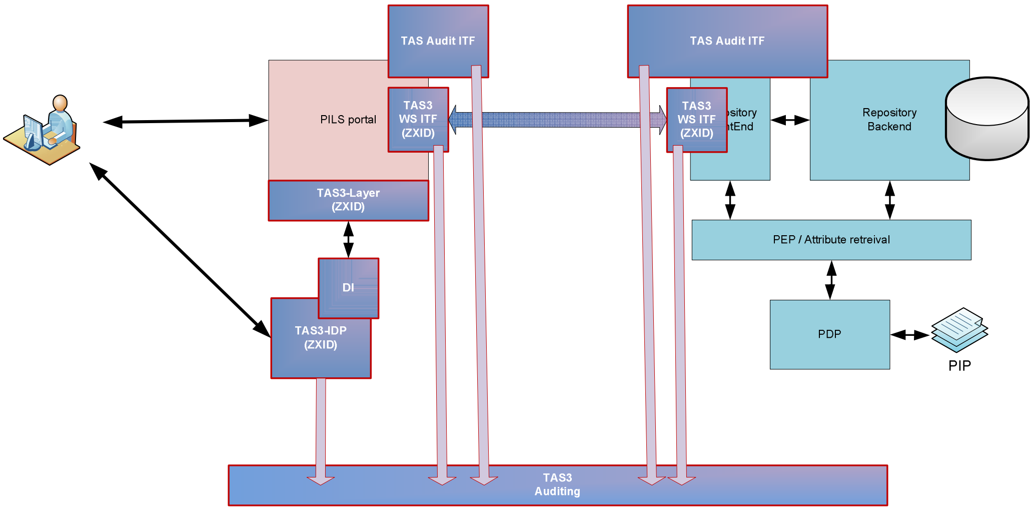


Figure 10: Phase 1 TAS³ component integration

Objectives

Achieving TAS³ compliance with the TAS³ communication protocol stack, i.e. TAS³ compliant SSO & SP communication (Figure 10).

(Secondary) Provide access log through TAS³ logging mechanism.

Success Indicators

Successful Integration means that the PILS/repository configuration remains functional, while relying on the above mentioned TAS³ components for establishing trust and security. Ease of integration (e.g. amount of glue code needed, architectural compatibility ...) and stability of the TAS³ components is to be considered an important performance indicator.

It should be possible to demonstrate the following parts:

- Login to PILS portal using TAS3-ZXID-IDP
    - SSO capability (Single SignOn)
    - SLO capability (Single Logout)
    - A functioning ID-FF Discovery Service
- PILS – Repository communication using the TAS3 communication stack.

The concrete trial scenario will be available at the time of demonstration (review) in a separate document.

The sequence diagrams in Figure 11 and Figure 12 give a high level overview illustrating the main TAS³ functionality that will be demonstrated in the healthcare trial and how it is integrated into the PILS/Repository legacy application. Figure 11 shows the interaction between a user (a physician in this case), his browser, the PILS portal application (website, deployed on a Tomcat Servlet container) and the TAS³ Identity Provider (independent service). Within the portal application itself, communication (integration) between the core application and the TAS3-ZXID Service Provider component (deployed as an independent Java Servlet) through the Servlet container session management mechanism is illustrated.

Figure 11 shows a successful logon to the PILS application using TAS³ Single Sign-On, in the case where no SSO session is yet active and a username/password authentication mechanism is used. The diagram annotations are considered self-explanatory.

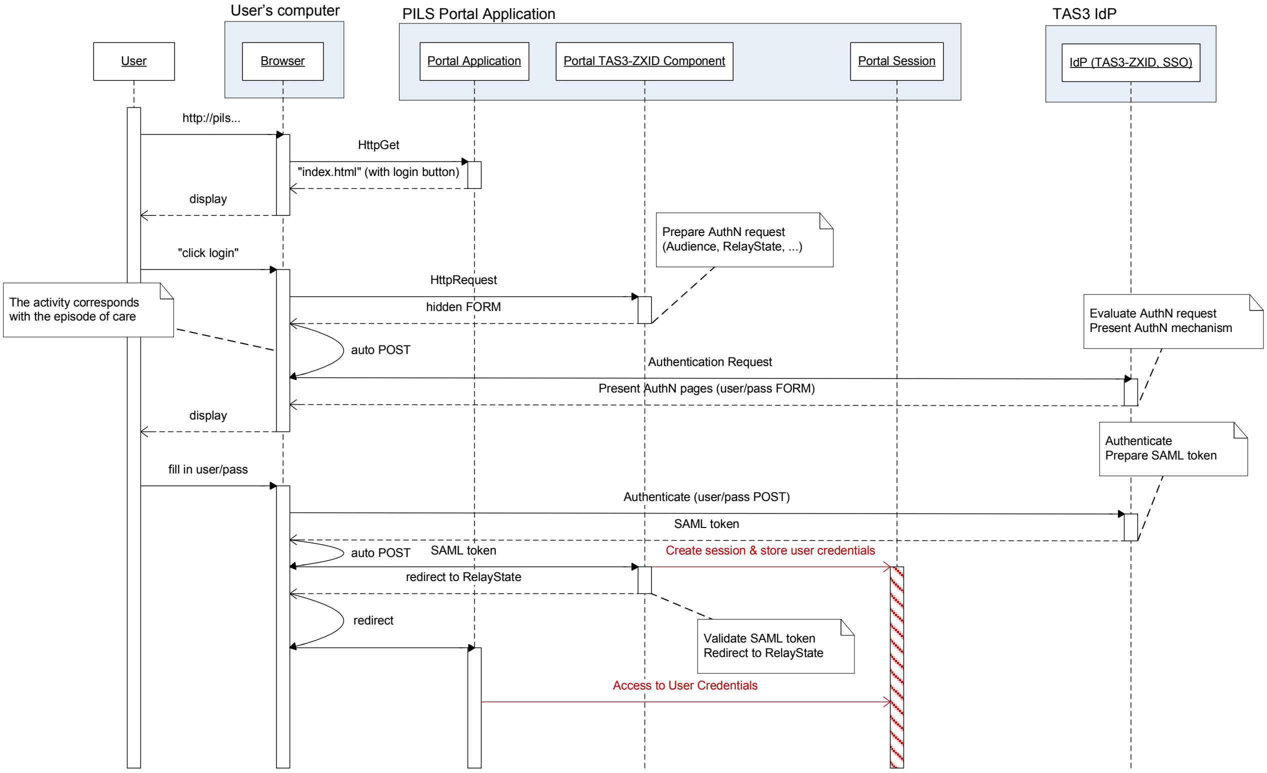


Figure 11: Successful SSO logon sequence diagram (SSO session not yet active, username/password authentication)

The integration of the TAS³-stack within the trial's legacy application is illustrated in Figure 12. The scenario is as follows: a logged in user wants to look up all available information on a specific patient. This results in following actions taken by the application:

- 'listRepositories', i.e. find all (trusted) repositories which can be queried for patient information.
- For each of the repositories found:
    - 'getDocumentsList' ask a repositories for information concerning a specific patient.

In the legacy implementation, the trust relationship between PILS and repositories is established statically. Repositories needed to be registered in the PILS application (together with their public key certificate for SSL mutual authentication). The Portal relies on this local configuration for the 'listRepositories' function.

TAS³ aims to provide trust management, therefore in the trial, trust between portal and repositories is established using TAS³ components, which means relying on the Liberty Circle-of-Trust mechanics. Trust is therefore managed at the IdP level (which offers obvious advantages), consequently the 'listRepositories' function which looks for available trusted repositories needs to rely on the TAS3 discovery services, which are encapsulated by the TAS³ stack.

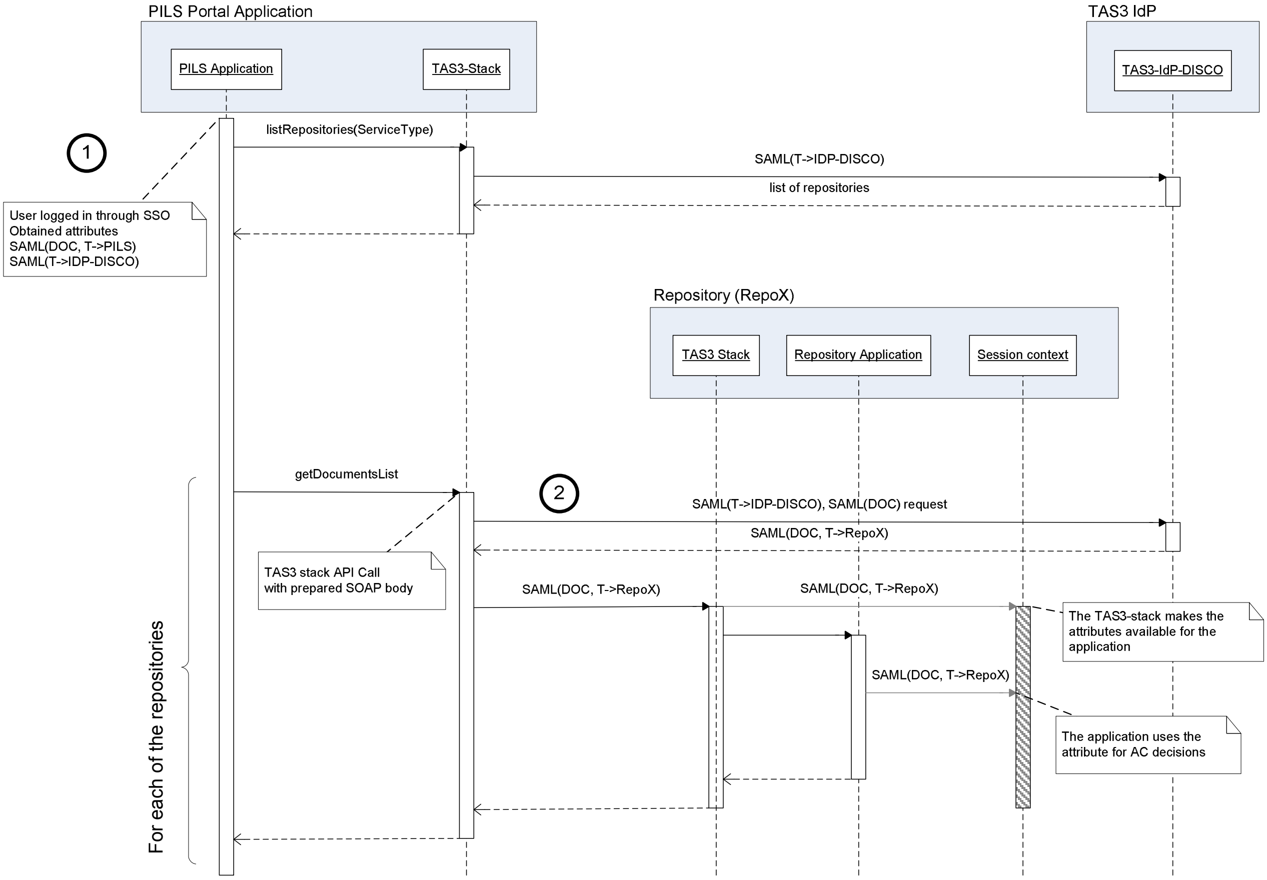


Figure 12: Service communication based on the TAS3 stack

Two aspects of Figure 12 require some explanation:

- ([1] on Figure 12) During the logon (TAS³ SSO) process, the IdP provides the portal application with assertions of the user: the two which are relevant here are (abstraction is taken from the encryption of the assertions, assertions are encrypted for their target):
    - SAML(DOC, T->PILS), a SAML credential targeted to the PILS proving that the user is a physician.
    - SAML(T->IDP-DISCO), the ID-WSF discovery bootstrap credential.
- ([2] on Figure 12) In order to properly authenticate the end-user (the portal acts on behalf of the user) with the repositories, the portal needs to retrieve appropriate credentials for each repository, i.e. a credential that proves that the user is a physician, targeted to the queried repository (SAML(DOC, T->RepoX)). This tedious job is automated by the TAS³ stack, and does not require additional programming. Note that this

correct credential translation was one of the issues not covered by the original legacy implementation .

3.5.3 Phase 2

While the first phase of the integration effort mainly concentrates on standards compliance and testing of basic TAS³ components, phase 2 will focus on demonstrating advanced functionality made available through TAS³.

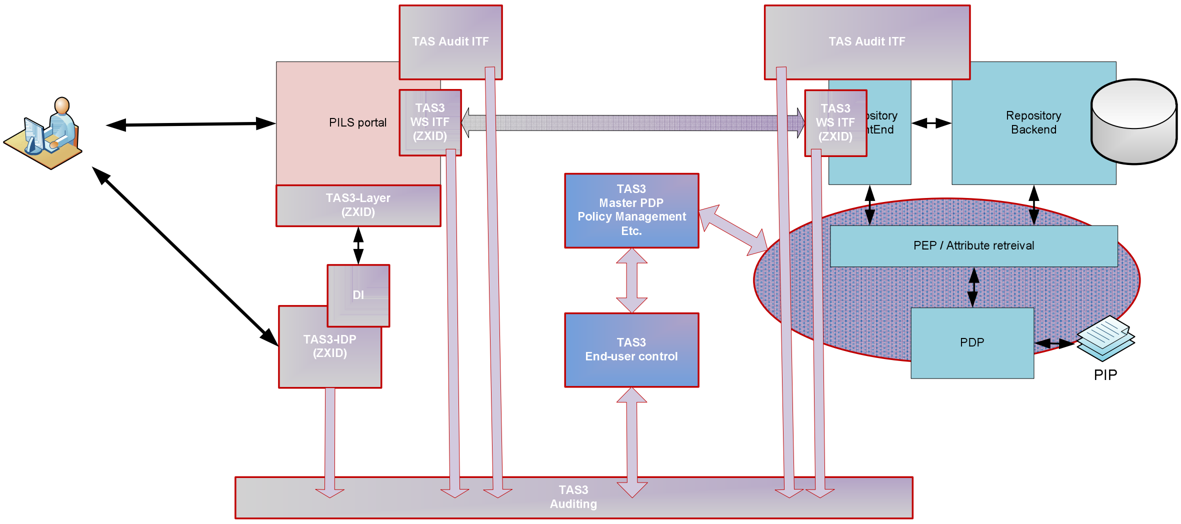


Figure 13: Phase 2 envisaged TAS³ component integration

The planning of phase 2 can only be finalised after the evaluation of the phase 1 results. Phase 2 will focus on the introduction of user-centricity in the management of personal information and the authorisation part of the architecture (see Figure 13). A large part of the current (legacy) implementation of the application access control is conceptually equal to the TAS³ approach, which will again allow for a gradual migration towards a full-blown TAS³ solution. The scope of the second integration phase and its technical implementation are heavily dependent on which TAS³ components which will be available at the time of planning.

4 Employability Integration Trial

4.1 Non-legacy application, ‘bottom-up’ development

4.1.1 Introduction

Student employability, as discussed in deliverables D1.1 and D1.4 as well as in the previous iteration of this document (D9.1) is a topic that has come under increased focus in the recent economic climate. As countries begin to see their way out of recession, it remains an issue for Higher Education in the UK and elsewhere that graduates want to see a return on their educational (and financial) investment and are seeking ways of improving their currency in the pressured market for jobs.

In the UK this is leading to an increase in activity in the area of student work placement. There are a myriad of such schemes, some associated directly with courses, others allowing students ‘time out’ from their degree. The associated complicated network of funding bodies, schemes and profiles is also difficult to manage, and many UK universities are employing the services of specialist placement providers, either internally or contracted from external sources, to widen and facilitate access by their students to such programmes.

To support the expanding demands of student placements this demonstrator looks at ways in which through using services the process can be both captured and automated as much as possible. Central to this development is the exchange of sensitive and private data between all parties: University, Student, Placement provider and companies offering employment. As the demonstrator develops beyond the initial integration trials into a full pilot situation, managing trust and privacy settings between all users of the system will increase in complexity.

Efficient flow and exchange of information about the students themselves, the programmes they are eligible for and the vacancies available is needed to support the matching of learners to appropriate placements. There are elements of choice at both ends of the process: learners want to be able to choose from a selection of suitable placements, while employers wish to choose from a selection of suitable students. Preservation of anonymity and gradual staged release of data, both from the students about themselves and the placement provider (employer) about the placement help to ensure fairness and impartiality throughout the process.

4.1.2 Choice as integration scenario

The scenario chosen for the integration trial is a thin slice of that outlined in the previous version of D9.1. It has been designed specifically to be flexible and focuses on showcasing the integration of TAS^3 technical components within the context of a a real-life situation.

Technically the demonstrator involves the integration of services with existing web-based sources of data and a few sources of legacy data. The central processes involved in the real life scenario are held largely within office procedures and are not automated, with much use of integrated computing applications. Thus in comparison to the Healthcare scenario the employability demonstrator operates in a more a bottom-up, 'green field' situation.

The lack of existing systems supporting the existing processes in the scenario has enabled us to focus on the user's needs in terms of planning user-centric service development and integration of the core TAS³ components. The development has followed the pattern of the creation of initial simple clients; their merger into more complex structures will increase in later phases of demonstration.

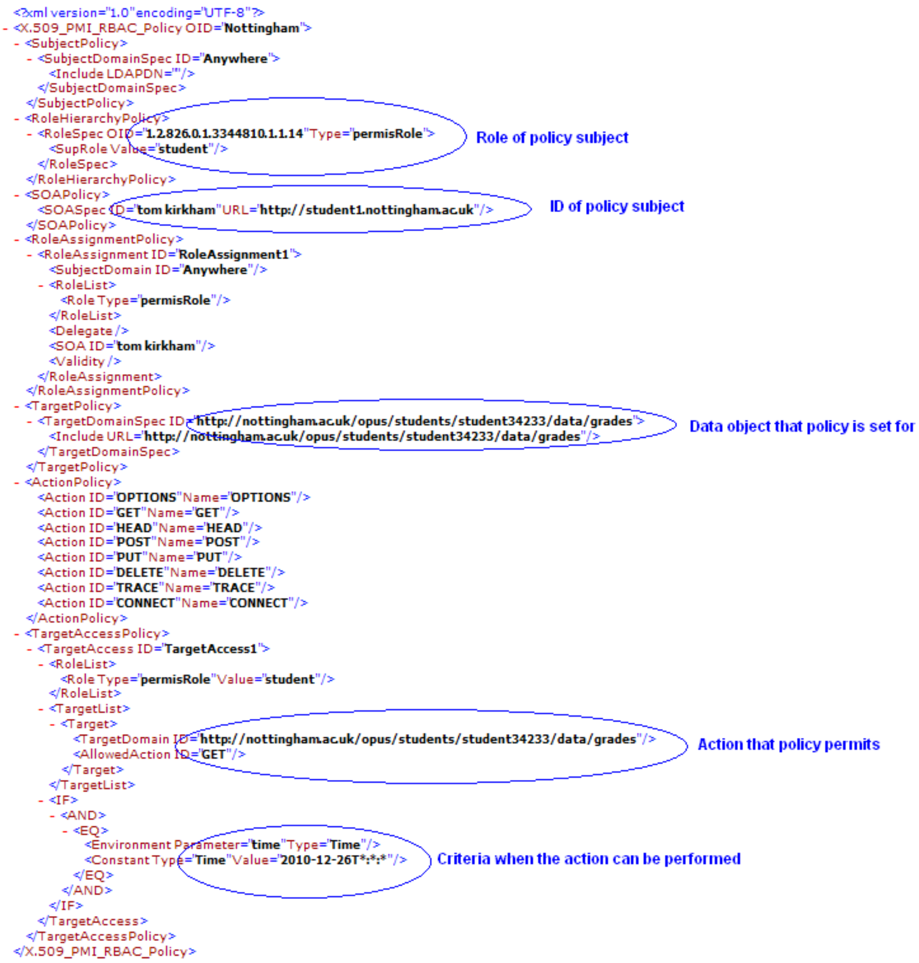
4.2 Application: description of use

4.2.1 Basic Storyboard

Prerequisites: the actors (placement administrators, placement providers and any application-specific service providers) are already registered with a TAS³ network and have agreed to any contractual obligations; vacancy profiles are available to the system, and have previously passed through a registration process similar to that for service providers.

1. A learner seeking a placement decides to register with a placement administrator who is able to offer places on a suitable programme. The learner is presented with the placement administrator system and is asked for authentication to ensure he or she is a genuine student (to ensure overall eligibility: most such programmes are only open to those who are current registered students with a UK Higher Education Institution). He or she is able to do this by selecting his or her preferred IDP and using SSO: if he or she is successfully authenticated he or she is able to log in.
2. If login is successful, a SAML token is returned which establishes the learner's institution and basic status (the course he or she is associated with). This is used locally to determine which programmes the learner is eligible for (e.g. only a learner on an Engineering course might be eligible for an Engineering programme) and he or she is presented with these in a list. The learner then selects the one(s) that he or she is interested in.
3. The learner is now asked to provide registration information specific to the programme (the system will only ask the user for information that is needed; this may vary according to the placement programme selected). This is the first point at which the user actively submits his or her data to the TAS³ infrastructure. It is envisaged that this data will be submitted via a registration form and will consist of links to external data stored in a variety of formats. In this demonstration we are using the EuroPass CV to represent this information (in later phases this will be substituted by ePortfolio data). This internationally-recognised format is used in over 70% of placements provided to students. In later versions of the demonstrator we will include other relevant forms of data, including ePortfolio data, which again can be linked to as separate data objects.
4. As the data is submitted the learner is asked to set details about how he or she wants his or her data to be used. This is done using policies: in the current integration trial the learner will be presented with a set of default policies from which to choose; for later pilots as the project progresses and components become available, users will be offered further options, first to

tune these and eventually to generate their own policies. In the current demonstrator these policies are created using PERMIS and enable the user to restrict access to specific data objects based on role, action and environmental factors such as time and date. A example of a PERMIS created policy can be seen in Figure 14.



```
  <?xml version="1.0"encoding="UTF-8"?>
- <X.509_PMI_RBAC_Policy OID="Nottingham">
  - <SubjectPolicy>
    - <SubjectDomainSpec ID="Anywhere">
        <Include LDAPDN=""/>
      </SubjectDomainSpec>
    </SubjectPolicy>
  - <RoleHierarchyPolicy>
    - <RoleSpec OID="1.2.826.0.1.3344810.1.1.14"Type="permisRole">
        <SupRole Value="student"/>
      </RoleSpec>
    </RoleHierarchyPolicy>
  - <SOAPolicy>
      <SOASpec ID="tom kirkham"URL="http://student1.nottingham.ac.uk"/>
    </SOAPolicy>
  - <RoleAssignmentPolicy>
    - <RoleAssignment ID="RoleAssignment1">
        <SubjectDomain ID="Anywhere"/>
      - <RoleList>
          <Role Type="permisRole"/>
        </RoleList>
        <Delegate />
        <SOA ID="tom kirkham"/>
        <Validity />
      </RoleAssignment>
    </RoleAssignmentPolicy>
  - <TargetPolicy>
    - <TargetDomainSpec ID="http://nottingham.ac.uk/opus/students/student34233/data/grades">
        <Include URL="http://nottingham.ac.uk/opus/students/student34233/data/grades"/>
      </TargetDomainSpec>
    </TargetPolicy>
  - <ActionPolicy>
      <Action ID="OPTIONS"Name="OPTIONS"/>
      <Action ID="GET"Name="GET"/>
      <Action ID="HEAD"Name="HEAD"/>
      <Action ID="POST"Name="POST"/>
      <Action ID="PUT"Name="PUT"/>
      <Action ID="DELETE"Name="DELETE"/>
      <Action ID="TRACE"Name="TRACE"/>
      <Action ID="CONNECT"Name="CONNECT"/>
    </ActionPolicy>
  - <TargetAccessPolicy>
    - <TargetAccess ID="TargetAccess1">
      - <RoleList>
          <Role Type="permisRole"Value="student"/>
        </RoleList>
      - <TargetList>
        - <Target>
            <TargetDomain ID="http://nottingham.ac.uk/opus/students/student34233/data/grades"/>
            <AllowedAction ID="GET"/>
          </Target>
        </TargetList>
      - <IF>
        - <AND>
          - <EQ>
              <Environment Parameter="time"Type="Time"/>
              <Constant Type="Time"Value="2010-12-26T*:*:*"/>
            </EQ>
          </AND>
        </IF>
      </TargetAccess>
    </TargetAccessPolicy>
</X.509_PMI_RBAC_Policy>
```

Role of policy subject

ID of policy subject

Data object that policy is set for

Action that policy permits

Criteria when the action can be performed

Figure 14: Example of a policy created using PERMIS

5. During the registration process the user also selects the trust criteria according to which his or her data should be handled. (In this iteration we will be using simple numeric trust metrics: later stages will develop this further to encompass the requirements identified in section 5.4 of the first iteration of D9.1. This identifies the need for trust in the information, in the system, in other parties, and full end-user control over information.) The trust metrics chosen here translate directly to the trust rankings of the service providers providing the matching service: at this stage of development trust metrics will rely on a simple calculation representing previous usage statistics of the matching services and associated user feedback. The policies selected by the user to secure the data and submitted at the registration form stage influence how Service Providers who could provide the matching service are selected.

6. Following the trust negotiation, the learner is presented with a list of suitable Service Providers: this can be none, one or many, depending on the criteria set earlier. (For current demonstration purposes we do not need to

limit this as we will be using dummy data; at a later stage we will consider an option to limit results if a very large number of matches is returned.) At this point the learner has the option to renegotiate the level of trust up or down in order to increase or decrease the number of results presented. The workflow needs to be able to handle issues that arise if none of the available services can live up to a user's high security settings, e.g. by reducing the level of trust the user needs to have in the SPs.

7. The learner selects and approves a matching service, and his or her CV data (and any other necessary criteria) are released to the service. The service is then executed and a list of anonymised matching vacancies returned to the learner. The execution process involves interaction of the main policy decision and enforcement points as data is exchanged outside of the TAS³ trusted domain to the external service provider supplying the matching service. This process can be seen in Figure 15.

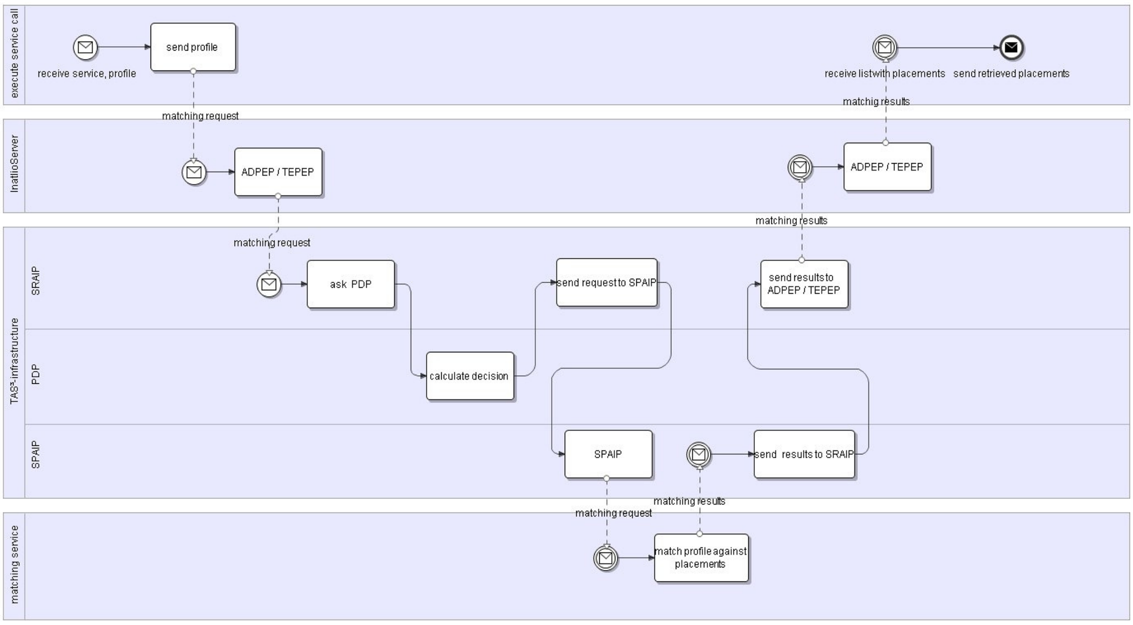
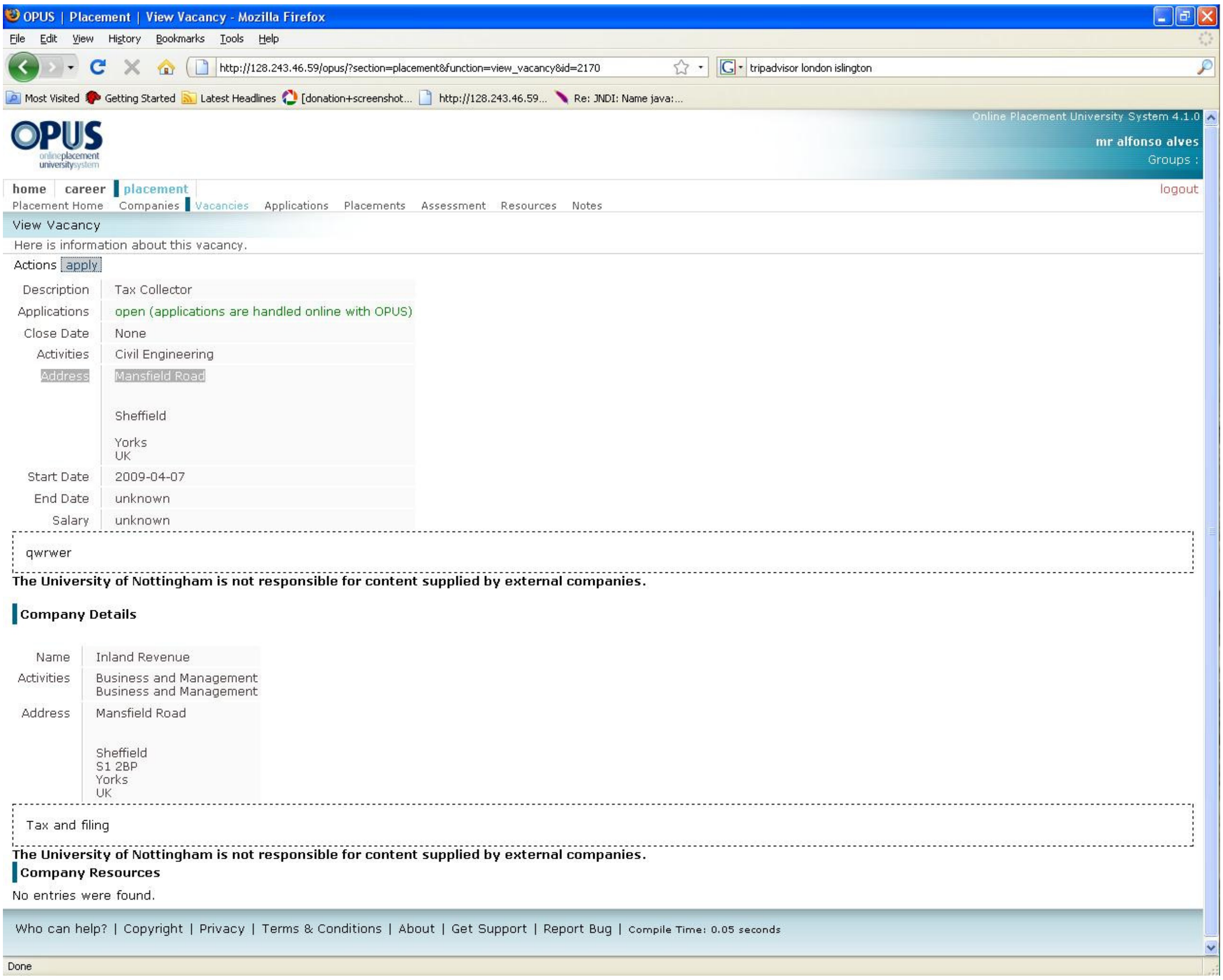


Figure 15: Intalio diagram showing business processes for the matching service

As the figure illustrates, the request to invoke the matching service is authorised, and this invocation is achieved via the policy framework. The results are also passed back through the policy management services to ensure security of the data. This phase of the demonstrator is a thin slice at this stage but is nevertheless able to illustrate how the policies are used within the system, how data is protected by the binding terms of the TAS³ infrastructure and how these can be expanded further to include service providers.

8. Once the match is complete the learner applies for the placements he or she is interested in and completes a more formal specific application, including a face-to-face interview. If the learner does not receive any matches or rejects the choices offered the whole matching process can be repeated with the learner changing his or her trust requirements.


OPUS | Placement | View Vacancy - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://128.243.46.59/opus/?section=placement&function=view_vacancy&id=2170
tripadvisor london islington
Most Visited Getting Started Latest Headlines [donation+screenshot... http://128.243.46.59... Re: JNDI: Name java:...
OPUS online placement university system
Online Placement University System 4.1.0
mr alfonso alves
Groups :
home career placement
logout
Placement Home Companies Vacancies Applications Placements Assessment Resources Notes
Done

View Vacancy

Here is information about this vacancy.

Actions apply

Description	Tax Collector
Applications	open (applications are handled online with OPUS)
Close Date	None
Activities	Civil Engineering
Address	Mansfield Road Sheffield Yorks UK
Start Date	2009-04-07
End Date	unknown
Salary	unknown

qwrwer

The University of Nottingham is not responsible for content supplied by external companies.

Company Details

Name	Inland Revenue
Activities	Business and Management Business and Management
Address	Mansfield Road Sheffield S1 2BP Yorks UK

Tax and filing

The University of Nottingham is not responsible for content supplied by external companies.

Company Resources

No entries were found.

Who can help? | Copyright | Privacy | Terms & Conditions | About | Get Support | Report Bug | Compile Time: 0.05 seconds

Figure 16: Vacancy match returned to user in OPUS

9. The TAS³ monitoring and auditing services are operational throughout. At the end of the process the learner is issued with a receipt that gives information about how her data has been used and the actions performed.
10. All data (apart from the audit data) associated with the transaction is destroyed: any new interaction with the system is via a separate workflow. The log events are never deleted, but these do not contain any PII.

Note:

- We recognise that this overall process has the potential to be extended to demonstrate handling of exceptions in future iterations:
- We could illustrate the user killing the process and revoking all rights to use shared personal data mid-flow
- We could also illustrate a service status changing and how it is renegotiated for a new service with an appropriate trust ranking
- We can alter the trustworthiness of a SP, so that it becomes selected or deselected accordingly
- We can alter the credentials of the user so some services become not available to him or her

- We can illustrate delegation in the model, as when a service provider needs more data than has been supplied by the learner to provide a match.

4.3 Integration Trial Objective

The integration trial objective is to illustrate a thin slice of the main TAS³ trust network functionality. This process has at its heart the user selecting and securing personal data for use in a specific application provided by a TAS³-compliant framework. The interaction in this framework will show how the policies are used in the system in order to protect the user, the service provider and the wider integrity of the application framework.

In practical terms, the integration of TAS³ systems and data is a key development challenge presented by the scenario. In terms of the wider scenario and real-life application, the demonstrator provides an insight into a new expectation of user interaction in terms of employability matching. The demonstrator will put the user at the centre of the process and in control of who provides the job matching, what personal data they can use, and how long thy can use it for.

This control by users over their personal data will from a technical perspective highlight the practical steps that TAS³ is presenting to show how user-centric privacy can exist in distributed computing applications. This privacy will extend in all directions from the demonstrator, allowing both employers and service providers to secure any sensitive information that they present to the system.

Overall, the following issues need to be addressed:

- The need to integrate systems and transfer data
- Provision of a better choice of matching facilities for learners
- Learners need to have control over their own data
- Privacy and anonymity need to be preserved, for the learner and the placement provider (employer)

In terms of the main aims of the demonstrator, the management of policies is at the centre of the work.

Further aims are to demonstrate:

- authentication/authorisation of access to sensitive data secured by both users and service providers
- policy setting/tuning managed in a user-friendly way but also secure enough for use in a complex and distributed system
- trust negotiation using metrics that the user can understand and relate to (negotiation must present meaningful results and ensure that the trust levels of service providers can be readjusted in real time, and in extreme cases re-negotiations can take place mid-process)
- some key service integration between application level services and the TAS³ trust infrastructure to demonstrate external service provider interaction with the trust framework

- that the user is at the centre and in control of use of personal data (achieved by feeding significant events in the process back to the user for approval; clear and understandable interfaces will need to be presented to the user, particularly for the policy-setting process and the key decision points in the workflow)
- a sizeable part of the complete integrated TAS^3 trust infrastructure in action (the flow will illustrate how a user can begin interaction with TAS^3 and continue through to the execution of an application in the framework and the retrieval of the result; this thin slice presents interactions with the key security and user specific components in TAS^3 and presents a strong basis for future developments).

4.4 Architecture

The diagram below shows how the architecture is built around the user. The user is put at the centre of the wider infrastructure by a series of interfaces that give direct access to the workflow and real-time notifications from the trust network. In order to explore how the architecture is being employed by the demonstrator, the diagram will be considered section by section.

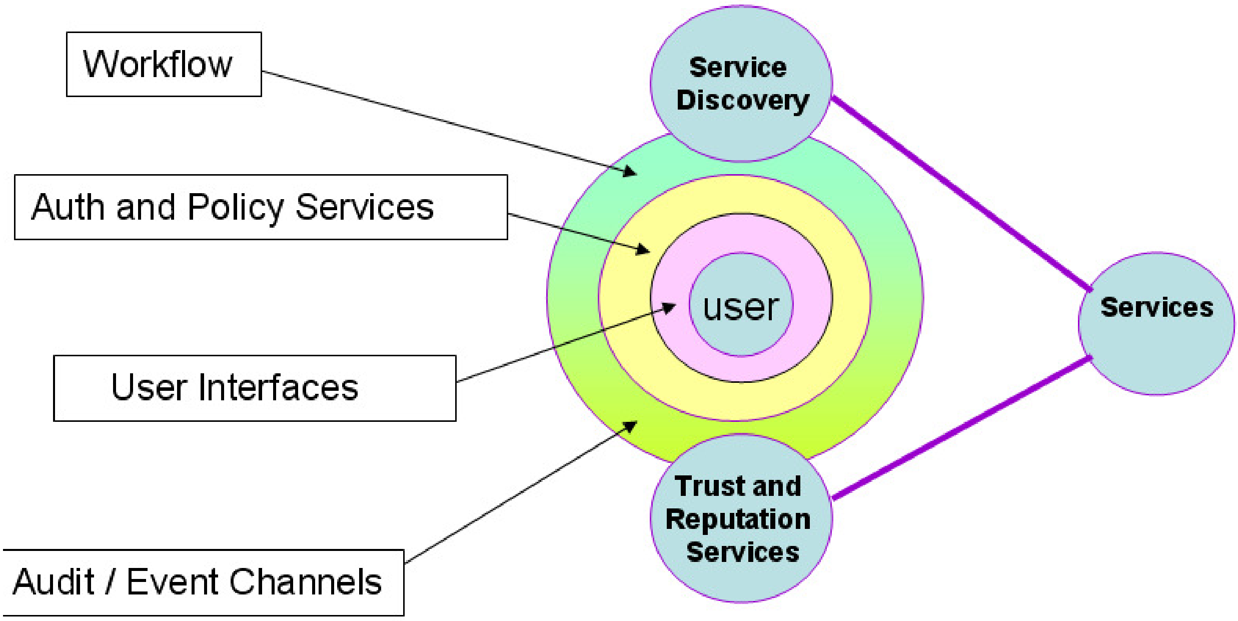


Figure 17: User at the centre of the architecture

The User

In the context of this demonstrator the ‘user’ is the learner seeking enhanced employability via a work placement. He or she will sign up for the TAS^3 application and present personal data to the services within the network to allow a search to be conducted. Future iterations of the model will also cover the service provider interaction as a user wishing to join the TAS^3-enabled network in order to publish services.

User Interfaces

The main user interface for this demonstrator is the Dashboard. We anticipate that the Dashboard will at a later stage be embedded within a web page or portal commonly used by the user (for example OPUS). The main function of the Dashboard will be to present the user with a point where interactions with the workflow and policy infrastructure can be made. For example, by using this channel the user will select programmes , set policies and specify acceptable trust levels.

Authorisation and Policy Services

These services will form the hidden layer of security within TAS^3. Here services will exist to enforce and make decisions based on policy associated with data and users' roles. In addition the services will provide validation of identification credentials, ensuring the system is secure and valid. The levels of assurance provided by these services will be translated in the quality of service guarantees that will be presented to users as they join.

Workflow

The workflow in the application ensures that the appropriate steps are carried out in the demonstrator. These steps are captured in BPEL format and are associated with endpoints of actual services following the service discovery, negotiation and selection processes. The workflow will constantly feedback data specific to the application for user approval.

Service Discovery

The service discovery service will maintain a repository of all members of the trust network and basic details of their usage criteria (i.e. policy). The workflow will select application specific service providers (i.e. matching service) via this service prior to workflow execution. If the services drop out of the workflow, the service discovery service will be invoked to find replacement services in real time.

Audit / Event Channels

Alongside the workflow monitoring of services, trust and reputation of appropriate service providers will be managed during the application execution. This is a vital to ensure that the application does not use services which could present trust or security breaches to the users. It is important to ensure that this monitoring data is linked to the sources of such security and trust data, for example logging of the authorisation and authentication services and trust framework.

In addition the audit event channels can be directed by the user via application settings to create specific logs in specific locations with regard to the application execution. This data can be made available to users for personal functions such as proof of access and other logging functions that may be needed after the entire application is completed.

Service (Matching Service)

The matching services will provide matches between user data and placement criteria and placement advertisements issued by companies. The demonstration service will compare the format of the vacancy used by the placement provider with user data held in EuroPass CV format[13]. The results of the match will be returned to the user as a result of the application invocation handled by the workflow.

The sequence diagram in Figure 18 shows how the components built by project partners will interact in this demonstrator.

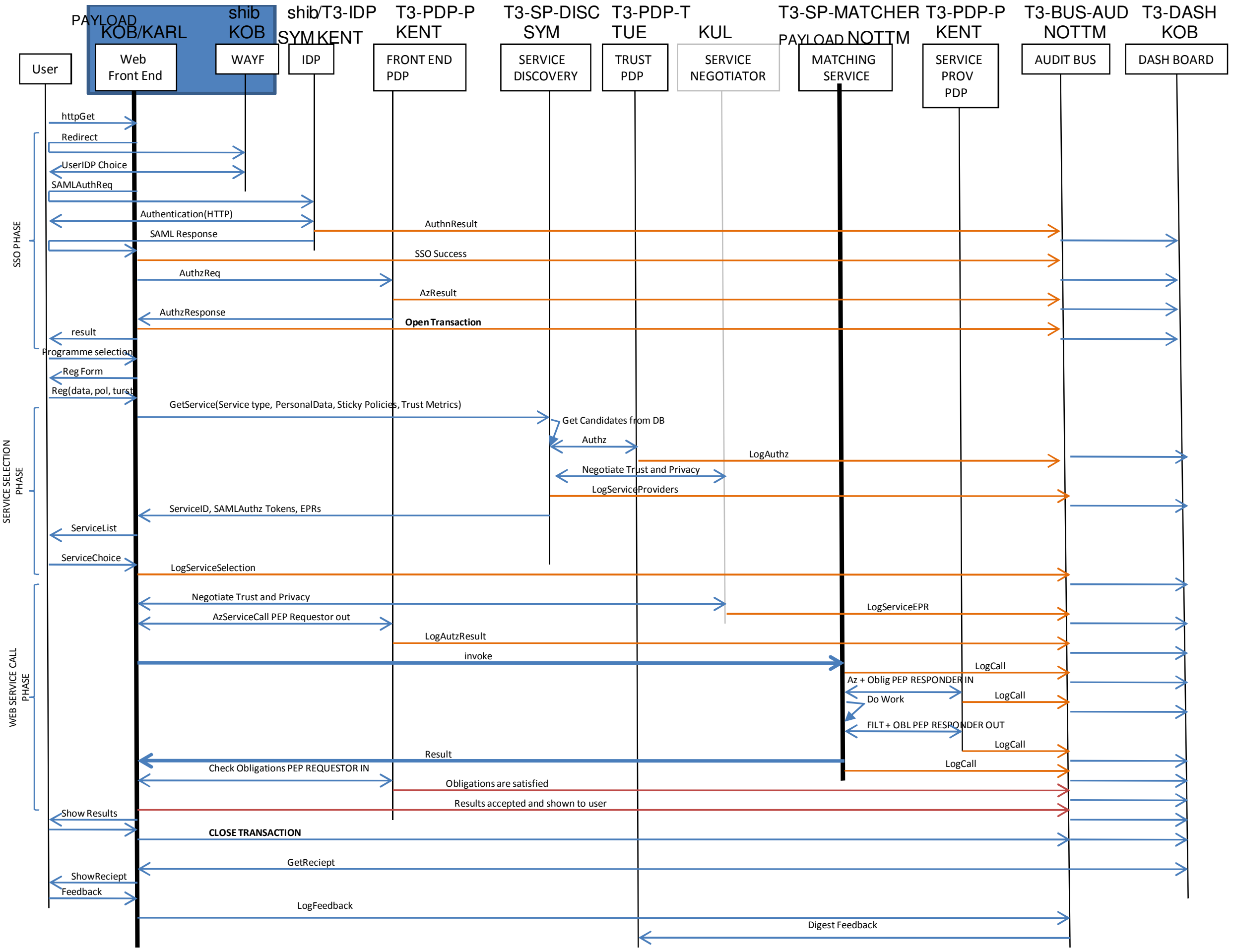


Figure 18: UK employability demonstrator sequence diagram

Overall the sequence is divided into three main phases. The first of these is the SSO phase, where the user requests, and is either granted or denied access to, the application that is running on the TAS³ framework. The user at this phase is represented by information held by their home organisation in SAML format. This basic limited data will be used to identify the role that the student has at that organisation (e.g. undergraduate student, postgraduate student, etc). Once granted access, the student will present personal data and select his or her data security policies in the service selection phase, SS.

[13] http://europass.cedefop.europa.eu/europass/home/vernav/Europasss+Documents/Europass+CV.csp

The SS phase will involve the user selecting a programme that he or she is determined to be eligible for based on the data given in the SSO phase. Once a programme is selected, the user is presented with a registration form which acts as an application form specific to that programme. As with a traditional application form, this will request personal data specific to the programme. This data can be input directly via a web form, or alternatively the user could give pointers to data held remotely (e.g. in an ePortfolio). As users are presenting personal data at this phase, this is the point where they select policies to protect it. This offers an improvement on traditional application methods as it offers finer granularity: for example different sections of a CV can be given different degrees of security. The SS phase then presents the user with a list of potential services to provide a job match based on the preferences and security/trust settings declared in their registration/application form. If no services can be presented at this point, the process can loop to allow the user to modify or add new security settings and try again.

The final phase is the actual call to get the placement match. This consists of the transfer of personal data across domains and involves obligations and policy checks. As in other phases, calls to policy enforcement and decision points and application-specific calls will all be logged and directed to the user Dashboard. The results of the match will be returned to the user; if the user accepts these and progresses to the next phase of application, the use case will be complete in this initial implementation. Finally a receipt will be passed to the user giving a summary of the Dashboard logging data with links to log sources for more detail.

The main achievements in this demonstrator are the integration of a variety of components generated by a number of partners. Technically the cross-domain management of user-secured data in the TAS^3 policy framework is an innovation that the project will build upon in further development phases. This is also reflected in the logging framework, with Dashboard interaction and the receipt as an additional log-based innovation, giving users the wherewithal to investigate application execution once it has completed, should they wish to do so.

4.5 Future Work

As with the healthcare work detailed in section 3.5 above, the work described here is for the first phase initial trial in which we will test the integration and interaction of available TAS^3 components in a realistic setting. While the healthcare demonstrator is trialling integration of TAS^3 components into an existing system, the employability demonstrator demonstrates how a new system can be developed using TAS^3 as a starting point.

Future work in phases two and three in the remaining two years of the project will extend this work into a pilot with users and service provider systems and focus on expanding the range of functionality in all areas.

While this first phase of the integration effort mainly concentrates on standards compliance and testing of basic TAS^3 components, phase 2 will focus on demonstrating the advanced functionality which becomes available as components develop and mature; it will also establish their connection to systems utilised by piloting service providers and users. This next phase will also focus on the introduction of user-centricity in the management of personal information and the authorisation part of the architecture (see Figure 13).

A final phase of development will extend this still further to incorporate further service providers, a wider range of end users and, if feasible, investigate application of TAS^3 to international data exchange involving the NL employability demonstrator. User uploaded data will be extended and developed from the summary Europass CV to include ePortfolio data collected by students using the Mahara system. In terms of storyboard, by this final phase the process will be demonstrated for a vacancy provider joining the system and securing vacancy information prior to it being matched. This will illustrate how other users can take part in the application, thereby bringing in user data from various sources and different groups.

As with the healthcare demonstration, the planning of phases 2 and 3 can only be finalised after the evaluation of the phase 1 results; the scope of the second integration phase and its technical implementation is heavily dependent on the TAS^3 components which will be available at the time of planning. In terms of user centricity, in both phases 2 and 3 testing and further development on the interfaces will take place following rigorous engagement with users. This involvement will often consist of monitoring, but in cases such as trust negotiation we expect to explore how interested users can influence the process in real time rather than using pre-set values.

To support this increased user centricity the flexibility of both the audit/event buses and workflow will need to be improved. As in the case of negotiation we will aim to make the user aware in real time of what is happening with the workflow and data in the system. This could be achieved in more detail with the development of interfaces to interpret both BPEL and logging data in more user-friendly formats.

In terms of technical development around policy and data security efforts to increase the level of technical integration between standards and software will continue. The work using ZXID and PERMIS is set to spearhead this.

5 Summary

This deliverable has described the two 'integration trials' designed and implemented within WP9, which embody the first practical (i.e. technical) steps towards establishing the TAS³ pilots. The description of the two trials shows that both application domains relevant to TAS³ WP9 are covered.

Within the healthcare domain the 'top-down' integration approach has been illustrated using the legacy PILS (Patient Information Location Service) application. In this trial, an existing security implementation is replaced by the TAS³ stack (aimed at offering a more elaborate and user-centric security functionality). The integration trial based on a scenario from the employability domain aims at establishing the advantages of using TAS³ in a 'green field' situation (i.e. a 'bottom-up' approach).

Both integration paths have been extensively described in this deliverable. They will be demonstrated during the March 2010 TAS³ review. At that point, detailed scripts of both demonstrations will be provided. The final outcome of the integration trials will be reported in 'D9.2 - Pilot evaluation report'.

6 Glossary

DAO	Data Access Object
GP	General Practitioner
GMD	Globaal Medisch Dossier (Belgian primary care health record)
HCP	HealthCare Professional
HIS	Hospital Information System
IdP	Identity Provider
KMEHR	Kind Messages for Electronic Healthcare Record
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PERMIS	PrivilEge and Role Management Infrastructure Standards
PHR	Personal Health Record
SAML	Security Assertion Markup Language
XACML	eXtensible Access Control Markup Language

Amendment History

Ver	Date	Autho	Description/Comments
2.0.1	2009-09-09	BC	eHealth Integration Trial Scenario / Doc outline start
2.0.2	2009-10-13	SEW	UK employability Integration Trial Scenario outline added
2.0.3	2009-10-27	TK	Additional demonstrator material added
2.0.5	2009-11-21	BC	Major Changes Healthcare part reflecting integration work done, placeholders new chapters
2.0.6	2009-12-01	BC	New introduction, healthcare updates
2.0.7-0.9	2009-12-16	BC,SEW	Integration of reviewer comments
2.1.0	2009-12-29	BC	Final Version